MASTER EN SCIENCES INFORMATIQUES

Etude d'un système générique d'aide à la gestion des accès

Vandurme, Alex; Voisin, Jean-Luc

Award date:
2001

Facultés Universitaires Notre-Dame de la Paix, Namur

Institut d'Informatique

Année académique 2000-2001

Étude d'un système générique d'aide à la gestion des accès

-

Alex Vandurme & Jean-Luc Voisin

Mémoire présenté en vue de l'obtention du grade de Licencié en Informatique

Résumé

Ce mémoire a pour principal objectif d'étudier la généricité de la problématique de gestion des accès. Pour ce faire, nous avons défini et appliqué une méthode qui permet, à partir d'avis recueillis lors d'une enquête, de concevoir un système d'aide à la gestion des accès (en abrégé SAGA) qui puisse s'adapter à de multiples domaines et qui inclut la prise en charge des demandes d'accès.
Vous ne trouverez cependant pas d'implémentation de SAGA mais bien la liste des classes d'objets et des concepts qui le composent, une découpe architecturale et quelques conseils pour son implémentation et sa mise en œuvre.

Mots clés :
- accès
- gestion des accès
- sécurité
- chemin d'accès
- multi-domaines
- générique
- demande d'accès
- méthode d'analyse

Abstract

The main purpose of this dissertation is to study the generic aspect of problems encountered in access management. From opinions collected in a survey, we have defined and applied a method that allows designing a system to help manage accesses. This system can be adapted to numerous fields and also deals with access requests. However, you will not find in this dissertation the implementation of this system but the concepts, the list of object classes, its architecture and some advice on how to implement it.

Keywords :
- access
- access management
- security
- access path
- multi-domains
- generic
- access demand, access request
- analysis method

Avant-propos

Nous tenons particulièrement à remercier :

Monsieur le professeur Jean Ramaekers, promoteur de ce mémoire, pour sa disponibilité et les conseils judicieux qu'il a prodigués tout au long de notre travail.

Les personnes qui ont collaboré à l'enquête : JF.Pirard, R.Debeire, JL.Bruwier, ML.Mackens, G.Ramelot, C.Lardinois, E.Sacré, C.Buffel.

Les membres du Jury pour l'intérêt qu'ils voudront bien porter à la lecture de ce document.

Toutes les personnes qui, par leur aide et leur soutien, nous ont aidés à aller au terme de ces études et à réaliser ce travail, en particulier notre famille.

Table des matières

Table des figures

Glossaire des termes et acronymes

Remarque : la définition précise des termes correspondant aux concepts du système étudié se trouve dans un chapitre spécifique (7. Définitions des concepts).

- *ActiveX* : ActiveX est un un petit morceau de code écrit dans le langage Java qui exécute généralement une tâche dans l'environnement du navigateur Web. Il s'agit en fait de l'équivalent Microsoft des applets Java.

- *Applet Java* : un applet est un petit morceau de code écrit dans le langage Java qui exécute généralement une tâche dans l'environnement du navigateur Web.

- *CA* : acronyme de Chemin d'Accès

- *CNF : acronyme pour Contrainte Non Fonctionnelle*

- *Coordinateur décentralisé* : il s'agit d'un gestionnaire d'accès qui ne fait pas partie de l'équipe centrale de gestion de la sécurité et qui, dans la plupart des cas, se trouve dans un autre bâtiment que celle-ci.

- *CORBA* : CORBA (Common Object Request Broker Architecture) est une architecture et des spécifications pour la création, la distribution et la gestion des objets dans un réseau. CORBA permet ainsi à des applications de communiquer entre elles.

- *DI* : acronyme de Demande d'Intervention

- *ERA* : acronyme signifiant Entity Relationship Association

- *FDA* : acronyme de Formulaire de Demande d'Accès

- *Firewall* : un firewall est un ensemble de programmes localisé sur un serveur qui sont destinés à protéger un réseau privé d'autres réseaux.

- *FTP* : FTP (File transfert Protocol) est un protocole logiciel destiné aux transferts de fichiers dans un environnement TCP/IP.

- *GRH* : il s'agit de l'abréviation de Gestion des Ressources Humaines. C'est le terme générique qui désigne la division ou le service des ressources humaines ; dans certaines organisations plus communément appelé service du personnel.

- *HTML* : HTML (Hypertext Markup Language) est un ensemble de balises de marquage inséré dans un fichier destiné à être afficher par un navigateur Web. Ces balises indiquent au navigateur comment formatter la page.

- *HTTP* : HTTP (The Hypertext Transfer Protocol) est un ensemble de règles pour échanger des fichiers (texte, images, son, vidéo, autres fichiers multimédia) sur le Web.

- *Impacter* : néologisme qui signifie « avoir des conséquences sur ».

- *Info Management* : application mainframe permettant de créer des enregistrements dont le texte est libre et qui peuvent être transmis à une autre personne. Ces enregistrements sont de trois types :
  - enregistrement d'activité : pour demander à un service de réaliser une activité pour un projet
  - enregistrement de problème : pour communiquer un problème et en faire le suivi de la résolution
  - enregistrement de projet : pour y stocker tous les enregistrements d'activités liés à un projet

- *LDAP* : LDAP (Lightweight Directory Access Protocol) est un protocole logiciel destiné à fournir des informations décrivant , par exemple, des organisations, des personnes ou d'autres ressources telles que des équipements dans un réseau.

- *Mainframe* : Nom donné aux ordinateurs centraux utilisant des terminaux pour la communication avec les utilisateurs.

- *Mémo formaté* : mémo est une ancienne application de messagerie électronique qui tourne sur mainframe IBM. Les mémos formatés sont des messages dont la structure respecte un modèle prédéfini et qui servent généralement de formulaire de demande.

- *Netware* : dans ce contexte-ci, il s'agit de l'ensemble des ressources de la plate-forme de gestion réseau de marque Netware

- *Open-Source* : le terme Open-Source fait référence à tout programme dont le code source est rendu disponible pour l'utilisation ou modification. Les applications Open-Source sont souvent développées par un ensemble de personnes et sont gratuitement disponibles.

- *Portefeuille d'application* : il s'agit d'une application informatique dans laquelle on recense toutes les applications informatiques qui existent dans l'organisation. On y reprend également la liste des personnes qui sont liées à chaque application ainsi que leur rôle vis à vis de celle-ci.

- *Proxy-HTTP* : serveur intermédiaire dans une communication client-serveur en protocole HTTP. Il agit en tant que serveur HTTP pour le navigateur Web et en tant que client vis à vis du serveur Web cible.

- *RAS* : abréviation pour Remote Access system. Il s'agit d'un système qui permet à des personnes de connecter à distance leur station de travail au réseau local, à l'aide d'un modem.

- *Record Infoman* : il s'agit d'un enregistrement de l'application Info Management (voir ci-dessus).

- *Ressources NT* : dans ce contexte-ci, il s'agit de l'ensemble des ressources de la plate-forme Windows™ NT.

- *RMI* : RMI (Remote Method Invocation) est l'équivalent Java de Corba.

- *RPC* : Remote Procedure Call (RPC) est un protocole qui permet à un programme d'exécuter des procédures présentes dans un autre programme localisé sur un autre ordinateur.

- *Security policies (Politique de sécurité)* : il s'agit de l'ensemble des règles de sécurité qui sont en vigueur dans l'organisation.

- *Servlet* : un servlet est un petit programme écrit en Java qui tourne sur un serveur.Web.

- *SMTP* : SMTP (Simple Mail Transfer Protocol) est un protocole utilisé pour envoyer et recevoir des courriers électroniques.

- *SNMP* : SNMP (Simple Network Management Protocol) est un protocole permettant l'administration et la surveillance d'équipement ou d'applications.

- *SOAP* : SOAP (Simple Object Access Protocol) est une méthode permettant à des programmes de communiquer entre eux même s'ils tournent sur des systèmes d'exploitation différents. Ils utilisent à cette fin le protocole HTTP et XML.

- *Subnet* : notion de sous-réseau dans le monde télécom

- *UML* : acronyme signifiant Unified Modeling Langage

- *URL* acronyme pour Unified Ressource Locator ; c'est une convention qui permet dans le domaine Internet de représenter les liens vers des informations.

- *Vax* : il s'agit de la plate-forme système de la firme Digital

- *WAPI* : acronyme pour Workflow Application Programming Interface

- *WorkFlow* : « On appelle workflow ou Gestion Électronique des Processus (GEP) la mise en œuvre explicite de processus inter applicatifs (sur les serveurs informatiques) comme de processus organisationnels (entre les intervenants), c'est-à-dire leur conduite après leur définition » [SACCONE, 1997].

- *XML* : XML (Extensible Markup Language) est une manière flexible de créer des formats communs d'informations et de partager le format ainsi crée et les données sur le Web.

1 Introduction

Il est indéniable que, de nos jours, un nombre croissant d'entreprises a pris conscience de la nécessité de protéger leurs ressources critiques par des systèmes de sécurité. Ces ressources sont de natures diverses. Il peut s'agir, par exemple, du fichier population d'une administration communale ou des locaux sécurisés d'une société bancaire. Lorsque l'on parle de sécurité, on pense immédiatement aux accès et à la gestion qui y est sous-jacente.

Au travers de ce document, nous allons montrer au lecteur notre démarche pour penser et concevoir un système informatisé centralisé d'aide à la gestion des accès. Toutefois, il existe déjà sur le marché des systèmes de gestion d'accès. Selon nous, ces systèmes ne permettent pas de gérer simultanément les accès dans tous les domaines présents au sein d'une même organisation. Nous pensons cependant qu'un système générique qui permette une gestion centralisée des accès aurait une valeur ajoutée indéniable pour l'organisation. Il nous paraît également important que ce nouveau système puisse s'intégrer dans l'environnement existant de l'organisation, s'adapter à la structure fonctionnelle de celle-ci et induire une simplification globale des procédures de travail. Nous pensons qu'un tel système devrait inclure la prise en charge de la problématique des demandes d'accès, sans toutefois oublier de laisser la place à la décision humaine.

Dans ce travail, nous souhaitons également mettre en évidence notre démarche d'analyse et de conception au travers d'une méthode qui, en partant d'une enquête, permet de produire un système d'information que l'on pourrait implémenter par la suite. Cette méthode accorde une attention toute particulière aux avis exprimés par les acteurs de terrain. Notre expérience et notre connaissance du domaine ont également pu être mises à contribution en apportant un éclairage complémentaire.

Pour ce faire, nous allons aborder la problématique de la gestion des accès par l'analyse rigoureuse des besoins et des procédures de travail existant dans trois domaines différents, à savoir les accès à des ressources informatiques, les accès à des locaux sécurisés et enfin les accès à des emplacements de stationnement dans le centre ville namurois.
Nous agrégerons ensuite sous forme d'une liste d'exigences l'information ainsi recueillie. Cette liste servira alors de point de départ à une analyse fonctionnelle dont le produit sera un modèle conceptuel des données et des scénarios de traitement. La dernière étape de ce processus consistera à concevoir, à partir de cette analyse, une architecture et des classes d'objets qui, une fois implémentés, formeraient un système complet d'aide à la gestion des accès qui satisfait aux exigences de départ. Le chapitre 4.'Méthode' décrit en détail la méthode utilisée.

Concevoir un système qui rencontre l'ensemble des exigences exprimées représente un travail considérable qui dépasse largement le cadre d'un mémoire universitaire. C'est pourquoi nous avons décidé de nous focaliser sur l'étude de la généricité de la problématique de gestion des accès. On pourrait résumer ceci par une double question : qu'y a-t-il de commun entre la gestion des accès aux coffres d'une banque, celle des accès aux emplacements d'un parking et celle des accès à un serveur de fichiers informatiques ? Comment peut-on concevoir un système unique qui soit utilisable dans tous ces domaines ?

2 Généralités

2.1 Conventions

Tous les *termes* soulignés en pointillé sont définis dans le glossaire qui se trouve en page 7.

Tout au long de ce document, nous désignerons par le terme SAGA - abréviation de « Système d'Aide à la Gestion des Accès » - le système que nous voulons concevoir.

2.2 Définitions de base

Les définitions suivantes ne constituent pas une liste exhaustive des concepts abordés dans le mémoire mais un ensemble de notions qu'il nous paraît important de préciser dès le début pour faciliter la compréhension du lecteur.

Il s'agit de définitions assez succinctes pour une première approche intuitive. Vous retrouverez les définitions complètes et précises de tous ces concepts dans le chapitre « 7 Définitions des concepts ».

- *Système*
  Dans ce travail, lorsque l'on parle de système, il faut entendre SAGA.
  Dans les autres cas, il est précisé de quel système il s'agit.
  Nous parlons également de système externe pour désigner un système d'information qui est externe à SAGA.

- *Ressource*
  Une ressource est une fonctionnalité ou un objet dont il peut être fait usage et auquel on peut demander accès.

  Exemples de ressources :
  - Plate-forme MVS,Unix,NT
  - Internet (Web,Mail,...)
  - Composant réseau (routeur, switch, firewall,...)
  - Application
  - Locaux, emplacement de parking

  Chaque ressource du système appartient à une personne ou à une entité organisationnelle que l'on appelle généralement *propriétaire de la ressource*. Chaque ressource du système est gérée par un *gestionnaire de ressources* (personne ou entité organisationnelle). Dans certains cas, mais pas tous, le gestionnaire de la ressource est aussi son propriétaire.

- *Service*
  Les ressources sont présentées aux utilisateurs bénéficiaires sous forme de services. Un service pouvant inclure une ou plusieurs ressources.

- *Accès*
  Un accès représente le droit d'utiliser une ressource. En toute généralité, on parle d'accès à un service pouvant représenter une ou plusieurs ressources. Il ne faut pas confondre avec l'utilisation de l'accès qui consiste en l'usage que l'on fait de ce droit.

- *Demande d'accès*
  Une demande d'accès définit l'action de requérir l'utilisation d'un ou plusieurs services. La nature de la demande d'accès consiste soit en la création d'un nouvel accès soit en la modification ou la suppression d'un accès existant.

- *Gestion des accès*
  La gestion des accès consiste à entreprendre toutes les actions nécessaires afin de tenir la liste des accès à jour. Elle inclut également la gestion des demandes d'accès (ajouter/modifier/supprimer un accès)

- *Gestion des demandes d'accès*
  Globalement, la gestion d'une demande d'accès consiste à mettre à disposition du demandeur une interface lui permettant de sélectionner le service auquel il désire avoir accès, ensuite à vérifier la légitimité de cette demande et enfin à transmettre celle-ci aux intervenants.

- *Bénéficiaire*
  Le bénéficiaire est une personne susceptible de tirer profit d'un ou plusieurs services auquel il a accès.

- *Utilisateur du système*
  Nous appelons utilisateur du système les personnes qui manipulent le système, qui posent des actions via le système.

- *Gestionnaire d'accès*
  Le gestionnaire d'accès est la personne chargée d'administrer les accès.

- *Gestionnaire de ressources*
  Le gestionnaire de ressource est la personne chargée de faire la gestion des accès. Il intervient pour configurer les ressources afin de mettre en œuvre une demande d'accès.

3 Objectifs et limites du système

3.1 Objectifs de SAGA

Le système que nous souhaitons mettre en place veut rencontrer les objectifs suivants :

1. être un outil d'aide à la gestion :

   Il sera tout d'abord un outil dont l'utilisateur fait ce qu'il veut. En effet, il n'est utile d'investir dans l'acquisition d'un outil que si on en a besoin et que s'il peut rendre des services. Le système devra donc être utile et utilisable. Ensuite, il s'agit d'une aide qui sera apportée aux différents acteurs qui vont utiliser le système. Celui-ci ne suppléera pas à la décision ni à l'action humaine mais donnera des moyens de mieux la poser. Il devra également permettre d'avoir une vue d'ensemble des accès, ressources et utilisateurs. Remarquons toutefois que le système se veut être une aide opérationnelle pour la gestion des accès et qu'en aucun cas il ne résoudra les problèmes organisationnels existants.

2. dégager des économies :

   Le système permettra de réduire les coûts de gestion, diminuera le nombre d'erreurs et dégagera des gains de temps. En outre, de par la centralisation de la gestion des accès, il permettra de dégager des économies d'échelle.

3. permettre une simplification :

   Le système simplifiera et allégera la tâche des différents acteurs (demandeur, gestionnaire d'accès, gestionnaire de ressources, ...). Ces personnes sont souvent des acteurs très spécialisés pour qui ni la gestion des accès et encore moins celle des demandes d'accès n'est considérée comme une tâche clé. Le système proposé ne devra pas alourdir leur tâche ou devenir une contrainte supplémentaire. Il doit idéalement leur apparaître comme un système d'aide dans l'exécution de leur tâche. En outre, il devra permettre l'automatisation de certaines tâches, être facile à gérer et à mettre en œuvre.

4. être performant :

   Le système doit être intuitif, rapide et convivial. Il doit être efficace dans le traitement des demandes et assurer une prise en charge systématique de celles-ci. Pour cela, il doit être disponible à tout moment et accessible à partir de tout endroit d'où les acteurs exercent leurs activités. Le système doit garantir la cohérence des informations.

5. s'intégrer et intégrer :

   Le système doit s'intégrer facilement dans la structure organisationnelle et avec l'infrastructure informatique existante, ce qui comprend les systèmes de contrôle d'accès existants. Il doit permettre non seulement la gestion des accès mais doit aussi prendre en charge la problématique des demandes. Le système devrait pouvoir s'adapter à la manière de travailler de l'organisation et pas, comme c'est souvent le cas avec d'autres logiciels, forcer l'entreprise à adapter ses procédures opérationnelles et son organisation au logiciel.

6. pouvoir évoluer :

   Le système doit être suffisamment modulaire que pour pouvoir s'adapter à l'évolution des besoins de l'organisation. Il doit garantir une bonne interopérabilité en étant ouvert aux standards.

7. être générique :

   Le système devra être générique et donc pouvoir s'adapter à de multiples domaines.

8. communiquer et faire communiquer :

   Le système permettra aux différents acteurs de communiquer efficacement entre eux. A ce titre, il devra permettre à chacun d'avoir « sa » représentation de la même réalité. Le système se positionne véritablement comme un acteur à part entière qui communique avec les autres acteurs par des messages et leur permet d'interagir avec lui. On peut également le voir comme une sorte « d'outil fédérateur ».

3.2 Limites de SAGA

Tout au long de l'analyse et la conception de SAGA, nous allons mettre en évidence des limites du système et des pistes de prolongation de celui-ci. Il nous semble cependant important de préciser d'emblée, afin qu'il n'y ait pas d'ambiguïté, que SAGA est un système qui aborde la problématique de gestion des accès sans aborder celle du contrôle de l'utilisation des accès.

4 Méthode

4.1 Introduction

L'objectif de ce chapitre est d'expliquer la méthode que nous avons utilisée pour d'une part recueillir les exigences et d'autre part concevoir un système d'information qui rencontre celles-ci. Pour ce faire, nous avons procédé en plusieurs étapes que l'on peut regrouper en deux grandes parties :

Partie 1 : l'ingénierie des exigences

- Étude de l'existant et analyse des besoins
- Synthèse des exigences

Partie 2 : la conception du système

- Modélisation des traitements
- Modélisation des données
- Découpe fonctionnelle
- Architecture
- Découpe en classes d'objets

Les sous-chapitres ci-dessous présentent la méthode et les objectifs liés à chacune de ces étapes.

La
Figure 1 - Méthode d'analyse' donne une vue globale de la méthode en faisant apparaître l'enchaînement des différentes étapes, les produits de celles-ci ainsi que les informations qui les alimentent.

Il est important de noter que pour l'ingénierie des exigences, nous avons travaillé en largeur, c'est à dire en abordant la problématique dans son entièreté.
Par contre, pour la conception du système, nous avons travaillé en profondeur en nous focalisant sur un seul scénario. Cette attitude est essentiellement guidée par le souci de présenter complètement la méthode, sans pour autant dépasser la taille d'un mémoire de fin d'études.

La découpe en classes d'objets constitue la dernière étape de notre travail, en effet nous avons choisi de ne pas réaliser d'implémentation dans le cadre du mémoire. Si la méthode exposée dans ce chapitre devait couvrir le cycle de vie complet de conception d'un logiciel, elle devrait comporter des étapes supplémentaires : notamment des étapes de vérification et de validation, des phases de déploiement, ...

4.2 Traçabilité

Tout au long de ce travail, nous avons accordé une attention particulière à la traçabilité. Nous entendons par traçabilité la possibilité de retrouver l'origine d'un choix ou d'une transformation. Nous suivons en ce sens les recommandations du Professeur Eric Dubois qui souligne qu'une des qualités attendue d'un cahier des charges est qu'il soit traçable « Un historique de chaque exigence est conservé et peut être remonté » [Dubois, 1998].

Cela se concrétise notamment par le tableau du chapitre 6.7 (Figure 4 - Tableau de traçabilité de l'étape de synthèse des exigences) qui explicite la traçabilité de l'étape de synthèse des exigences. On retrouve également la traçabilité au niveau de chaque fonction pour laquelle on mentionne les exigences qu'elle satisfait.

4.3 Schéma global de la méthode

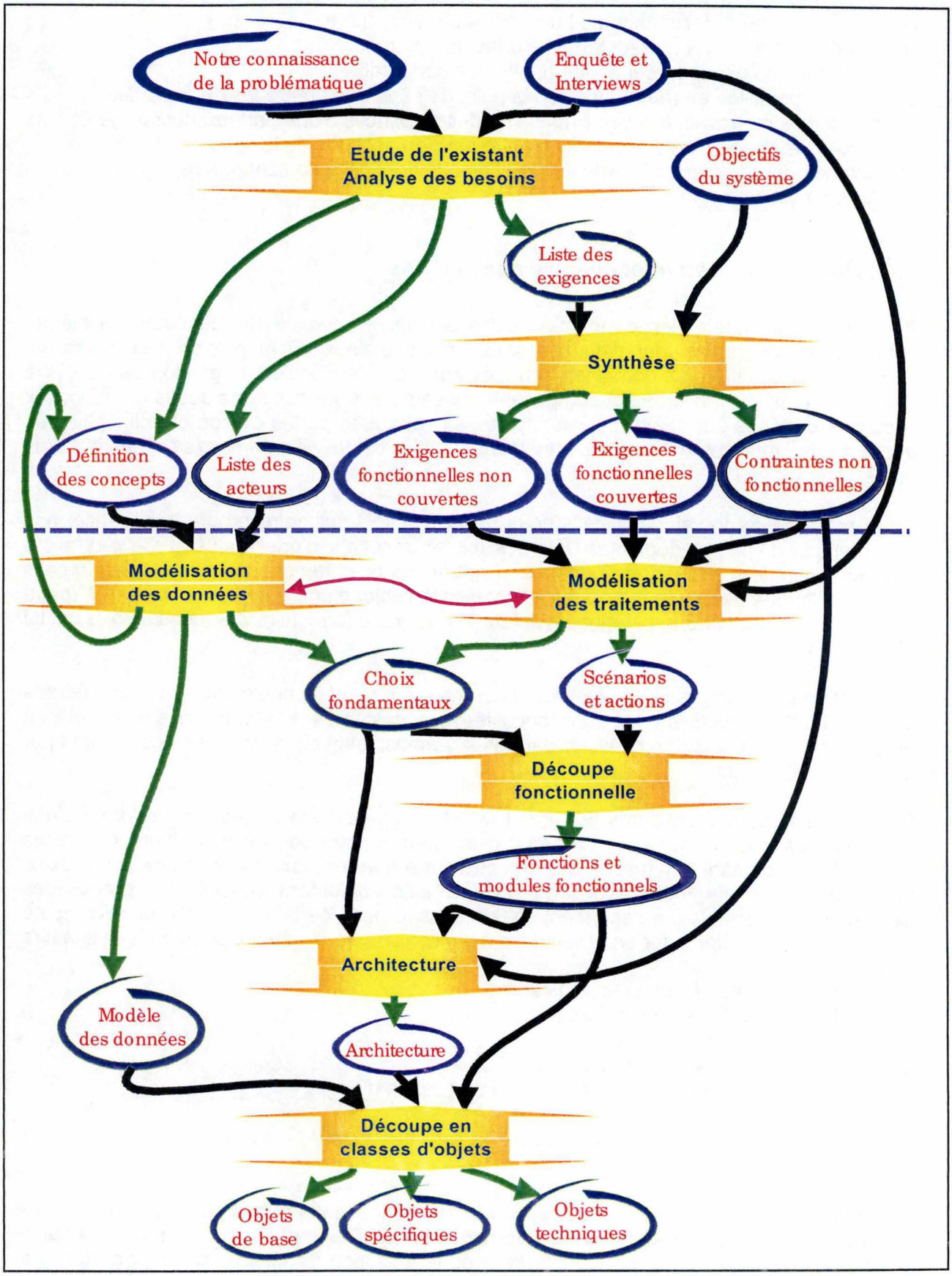

Figure 1 - Méthode d'analyse

Légende :

- Les étapes sont représentées par des 
- Les sources d'information et les produits des étapes sont représentés par des 
- Les flèches en **noir** représentent une relation « alimente »
- Les flèches en **vert** représentent une relation « produit »
- La flèche en **rose** représente une relation de coopération
- La ligne pointillée en **mauve** indique le point de séparation entre les deux parties principales du travail, à savoir l'ingénierie des exigences d'une part (au dessus de la ligne) et la conception du système d'autre part (en dessous de la ligne).
- Notons que, par souci de clarté, les relations de traçabilité ne sont pas représentées dans ce schéma

4.4 Étude de l'existant et analyse des besoins

Afin de pouvoir déterminer les fonctionnalités principales que devrait offrir notre système, nous faisons un rapide tour d'horizon dans trois domaines où la gestion des accès est primordiale. Les différents domaines analysés sont, en premier lieu, la gestion des accès à des ressources informatiques et composants réseaux, ensuite celle des accès à des locaux sécurisés protégés par des dispositifs à carte, sas et autres portes à code et enfin celle des accès à des emplacements de stationnement dans le centre ville namurois (cartes d'accès, bornes amovibles,...).

Notre méthode de travail consiste à nous baser à la fois sur notre propre expérience, nos observations dans ces domaines et enfin sur les résultats d'une enquête. Celle-ci a été réalisée sous forme de questionnaires distribués aux différents acteurs des domaines précités. Chaque personne interrogée a fait ensuite l'objet d'une entrevue. L'annexe 1 (page 144) contient pour chaque catégorie d'acteur interrogé, le formulaire du questionnaire qui lui correspond.

Le questionnaire comprend deux volets : le premier a pour objectif d'analyser les procédures de travail et de découvrir les fonctionnalités des systèmes existants, le second vise à déterminer les besoins non couverts par ces systèmes, lesquels sont jugés nécessaires par les différents acteurs.

Nous présentons aux chapitres 5.2 et 5.4 les résultats de cette enquête sous forme d'une synthèse des interviews. Celle-ci reprend d'une part le point de vue des différents acteurs interrogés concernant la situation actuelle et d'autre part les besoins exprimés par ceux-ci. Dans cette synthèse, nous avons reproduit le plus fidèlement possible les propos des personnes interrogées. Il a cependant été nécessaire dans certains cas, afin de lever toute ambiguïté, de les reformuler en utilisant les définitions de base développées dans le chapitre 2.2.
Les différents points de vue présentés sont :

- celui des utilisateurs /demandeurs
- celui des gestionnaires d'accès
- celui des gestionnaires de ressources
- celui d'un développeur d'une application de gestion d'accès

Vous trouverez également au chapitre 5.3 la liste des acteurs du système qui ont pu être identifiés.

Grâce à l'enquête et à notre connaissance du domaine, nous avons pu établir la liste des concepts du système d'information. Ce dernier point fait l'objet d'un chapitre à part (chapitre 7 – page 49) car la liste a été complétée avec la définition de chaque concept au fur et à mesure que nous avons opéré des choix au moment de l'étape de modélisation des données.

4.5 Synthèse des exigences

L'étape de synthèse des exigences consiste à déterminer, sur base des exigences émises par les différents acteurs et en respectant les objectifs que nous nous sommes fixés, les exigences qui seront couvertes par notre système et celles qui ne le seront pas. Certaines de ces exigences correspondent à des contraintes non fonctionnelles. Les exigences fonctionnelles non couvertes ont été écartées parce que nous avons jugé qu'elles ne correspondaient pas aux objectifs de départ.

Notons qu'au cours de l'analyse du système, cette liste a été revue. Certaines exigences qui se sont avérées trop complexes à couvrir dans le cadre de ce mémoire ont été transférées vers la liste des exigences fonctionnelles non couvertes. Dans une situation réelle, ce choix aurait du être validé par le maître d'ouvrage.

4.6 Analyse du système

L'objectif de l'étape d'analyse du système est de définir le système qui va pouvoir répondre aux exigences émises. Cette étape comporte deux grands aspects – les données et les traitements – qui s'influencent mutuellement. C'est la raison pour laquelle nous avons réalisé de front la modélisation des données et la modélisation des traitements. Pendant cette activité, nous avons été amenés à faire de nombreux choix. Nous présentons ceux qui nous paraissent les plus importants dans le chapitre « 8.3 'Discussion des choix fondamentaux' ».
La modélisation des traitements produit un délivrable intermédiaire – les scénarios et actions – qui sera utilisé lors de la découpe fonctionnelle pour en déduire les modules fonctionnels du système.

4.6.1 Modélisation des données

La modélisation de données consiste à formaliser la structure des données ainsi que les relations qui existent entre celles-ci. Nous avons choisi d'utiliser à cet effet le formalisme *ERA* (Entity Relationship Association).
Remarquons que nous avons utilisé la liste des exigences afin d'opérer a posteriori une vérification du modèle des données.

4.6.2 Modélisation des traitements

Lors de cette étape de modélisation des traitements, nous avons déterminé les scénarios de traitement qui découlent des exigences fonctionnelles exprimées précédemment. Pour chacun de ces scénarios nous avons déterminé les différentes actions qui le composent ainsi que l'enchaînement entre celles-ci. Nous avons choisi la représentation *UML* des diagrammes de séquencement pour formaliser les scénarios.

4.6.3 Découpe fonctionnelle

La découpe fonctionnelle consiste à traduire les scénarios et leurs actions en fonctions du futur logiciel. Les fonctions techniques et utilitaires apparaissent lors de cette étape.
Ces fonctions sont ensuite regroupées en modules fonctionnels suivant différents critères : par les données qu'elles manipulent, par leur utilité dans le système, par leur capacité de réutilisation, ...
Selon nous, il s'agit typiquement d'une étape de créativité pour laquelle il n'y a pas véritablement de règle ou de mécanisme automatique.

4.7 Conception du système

4.7.1 Architecture

L'étape d'architecture consiste à déterminer la manière dont vont s'agencer les différents modules fonctionnels dans le futur logiciel. Il ne s'agit pas d'une architecture technique dans laquelle on mentionne des choix techniques mais plutôt d'une architecture qui propose des lignes de conduite pour l'implémentation. La stratégie de conception de l'architecture est basée sur la « découpe sur base des fonctions » et sur la « découpe sur base de la distribution » [Habra, 1999]. La découpe architecturale tient compte des contraintes non fonctionnelles.

4.7.2 Découpe en classes d'objets

Comme son nom l'indique, la découpe en classes d'objets consiste à identifier les objets du futur système, leurs attributs et leurs méthodes.
Nous nous sommes basés sur le modèle des données pour déterminer les classes d'objets de base de notre système; chaque type d'entité devient une classe d'objet.
Ensuite nous passons en revue la liste des modules fonctionnels pour ajouter de nouvelles classes d'objets ou des méthodes que l'on assignera à des classes d'objets existantes. Nous avons également tenu compte de l'architecture lors cette découpe.

5 Étude de l'existant et analyse des besoins

5.1 Introduction

Dans ce chapitre, nous synthétisons les informations que nous avons recueillies au cours de notre enquête.
Nous présentons ces résultats en les regroupant par point de vue :

- Point de vue des utilisateurs /demandeurs
- Point de vue des gestionnaires d'accès
- Point de vue des gestionnaires de ressources
- Point de vue d'un développeur d'une application de gestion d'accès

L'enquête nous a également permis de déterminer la liste des acteurs du système d'informations ainsi que la liste de tous les concepts. Ce dernier point fait l'objet d'un chapitre à part (chapitre 7 – page 49) car la liste a été complétée avec la définition de chaque concept.

5.2 Étude de l'existant (synthèse des interviews)

5.2.1 Point de vue des utilisateurs /demandeurs

- Jean-François est développeur informatique dans une organisation bancaire. Il a régulièrement l'occasion de demander accès à différentes applications et services informatiques pour pouvoir réaliser ses tâches-clé. Dans cette entreprise, les développeurs n'ont pas accès à l'environnement de production (données des clients) sauf lorsqu'il faut y intervenir pour résoudre un problème. Dans son domaine, Jean-François a donc l'occasion de faire deux types de demandes dont le traitement diffère :

  - Les demandes classiques à des ressources informatiques (exemple : accès à un serveur *FTP*, à un système de DB, ...).
    Jean-François introduit ce type de demandes par différents moyens qui ne sont pas uniformisés : mail, *Mémo formaté* (formulaire de messagerie), téléphone, via le progiciel *Info Management* (application mainframe permettant de créer des enregistrements dont le texte est libre et qui peuvent être transmis à une autre personne).
    La plupart du temps, les demandes doivent être approuvées par un supérieur hiérarchique par lequel la demande transite.
    Dans la plupart des cas, ce type d'accès n'est actif que le lendemain de la mise en œuvre de la demande. Le délai de prise en charge par le gestionnaire d'accès est variable. Dans certains cas, le gestionnaire d'accès exige qu'il y ait un délai suffisant entre le moment de l'introduction de la demande et la date de prise d'effet de cette demande.

  - Les demandes urgentes pour résoudre des problèmes ponctuels.
    Des procédures d'urgence existent pour permettre aux développeurs de demander un accès immédiat à certaines services prédéfinis (liste restreinte).
    Il s'agit des services dont ils ont normalement besoin pour intervenir dans l'environnement de production afin de résoudre un problème ponctuel important. Ces accès sont accordés pour une durée maximale de 2 jours et doivent faire référence à un enregistrement dans le progiciel *Info Management* où l'intervenant décrit le problème ainsi que les différentes interventions qui seront réalisées pour le résoudre. Ce référencement est le seul contrôle de légitimité qui est réalisé en plus du contrôle avec le *portefeuille d'application*. L'intervenant doit donc faire partie des personnes qui gèrent l'application en question.

Dans ce cas, l'effet de la demande (c'est à dire l'octroi de l'accès) est immédiat ; il s'agit donc d'une procédure automatisée qui réalise directement l'intervention dans le système de gestion des ressources, sans intervention humaine.

De manière générale, les accès accordés à Jean-François sont temporaires (c'est à dire pour une durée limitée). Dans certains cas, par exemple lorsqu'il s'agit d'une ressource dont il assure la maintenance, l'accès est donné pour une durée illimitée. Il faut également remarquer que dans ce domaine, tous les accès ne sont pas gérés par des services spécialisés (services de sécurité) mais parfois directement par le propriétaire de la ressource.

Jean-François pense que certains gestionnaires d'accès traitent les demandes d'accès par lot, spécialement quand la gestion des accès n'est pas leur tâche principale.

Il fait également remarquer que la prise en charge de la demande n'est que rarement immédiate et qu'il n'est pas souvent averti de celle-ci. C'est une fois la demande réalisée ou refusée que Jean-François reçoit un signal de prise en charge de celle-ci. Jean-François se plaint également de ne pas être averti lorsque le traitement de sa demande est terminé. Il effectue donc le suivi en téléphonant au gestionnaire d'accès ou en réitérant des essais de l'accès jusqu'à ce qu'il fonctionne.

Les demandes concernant plusieurs personnes ou ressources doivent généralement être introduites séparément.

Selon Jean-François, le contrôle de la demande d'accès se fait par rapport au rôle fonctionnel du bénéficiaire, à la nature de l'accès et surtout à la justification de la demande. Remarquons que, lors de ce contrôle effectué par le gestionnaire d'accès, il arrive que la demande soit refusée et qu'une autre solution soit proposée par celui-ci.

Remarque importante : nous n'avons pas jugé nécessaire de recueillir d'autres témoignages d'utilisateurs dans la mesure où nous disposons tous deux d'une large expérience en la matière.

5.2.2 Point de vue des gestionnaires d'accès

- Richard gère les accès aux différents locaux d'une société bancaire. Celle-ci comporte environ 120 portes et sas sécurisés.
  En général, la demande d'accès à un local lui parvient sous forme d'un courrier électronique émis par le supérieur hiérarchique de l'utilisateur.
  Richard, après avoir vérifié la légitimité de la demande en contrôlant que l'utilisateur est bien sous la responsabilité du demandeur et que l'accès demandé est nécessaire à la fonction de l'utilisateur, introduit dans l'application de gestion des accès Honeywell© les informations nécessaires à cet accès. Remarquons que Richard détermine lui-même, sans aucun système d'aide les impacts de chaque demande d'accès. Il envoie ensuite un courrier électronique à l'utilisateur afin de lui communiquer son code d'accès. Si la demande lui est refusée, il l'informe du motif de ce refus. Dans tous les cas, il informe également le responsable hiérarchique du demandeur de la suite donnée à la demande.

  L'application de gestion des accès Honeywell© importe quotidiennement la liste des utilisateurs provenant du service de gestion des ressources humaines. Elle permet de constituer des groupes d'utilisateurs, ce qui simplifie la gestion.
  Les ressources (portes et sas sécurisés), ainsi que les chemins d'accès sont encodés par ses soins. Il existe bien un plan graphique de ceux-ci mais aucun lien n'existe entre ce plan et le système de gestion d'accès.

  Points forts du système existant

  - Intégration avec l'application de la *GRH* ; y compris la suppression des accès lors d'un départ ou suite à une mutation.
  - Interface graphique de l'application très conviviale.
  - Possibilité de requêtes diverses sur les accès existants.
  - Gestion les groupes d'utilisateurs (mais pas en synchronisation avec la *GRH* car la notion de groupe n'est pas identique)
  - Alarming et monitoring.
  - Notion d' «accès de base » : ensemble d'accès accordés par défaut systématiquement et immédiatement (sans intervention) en fonction des catégories d'utilisateurs.
  - Accès limités dans le temps (période de validité).
  - Possibilité de déléguer la gestion à des correspondants décentralisés en leur limitant le champ d'action
  - Intervention immédiate sur les composants.

  Points faibles du système existant

  - Pas de traitement automatisé de la demande d'accès.
  - Archivage papier des demandes d'accès par ordre chronologique => faible capacité de recherche.
  - Maintenance de l'application très coûteuse.
  - Détermination manuelle (par le gestionnaire d'accès) des composants *impactés*.
  - Pas de possibilité de définir un accès qui sera activé dans le futur.
  - Pas d'importation des ressources ; gestion manuelle.
  - Pas de garantie du délai de traitement d'une demande.

- Jean-Luc est responsable de la sécurité informatique des petits systèmes (NT, UNIX) et également de la plate-forme d'accès sécurisée qui relie le réseau interne de la société au monde extérieur (Internet).

  Actuellement, les demandes d'accès lui parviennent le plus souvent oralement et par courrier électronique. Seules les demandes d'accès à Internet et au *RAS* (Remote Access System) font l'objet d'un traitement automatisé via une application développée en interne. Cette application est présentée au point 5.2.4 au travers d'une entrevue avec son concepteur.

  Dans les autres cas, il s'agit de demandes d'accès complexes qui sont traitées manuellement au cas par cas. Jean-Luc en contrôle alors la légitimité (analyse de risques, conformité aux *Security policies*), celle-ci est analysée ensuite dans le but de déterminer les ressources impliquées. Les responsables des ressources sont alors contactés pour vérifier la faisabilité de cette demande d'accès. Dans la plupart des cas, le demandeur doit s'inquiéter de l'état de la demande car le retour d'informations n'est pas systématique.

  Selon Jean-Luc, il est important que le système puisse se prononcer le plus tôt possible sur le caractère réaliste et réalisable d'une demande.

  Points forts du système existant

  - Analyse individuelle de chaque demande => meilleure sécurité mais coût élevé car fait par un humain.

  Points faibles du système existant

  - Pas de traitement automatisé de la demande d'accès.
  - Pas d'archivage des demandes d'accès.
  - Pas de retour systématique d'informations vers le demandeur.

- Marie-Louise est responsable de l'ensemble des accès aux ressources IT dans une banque. Ces ressources comprennent principalement le *mainframe*, les plates-formes NT et UNIX, les accès Internet et la messagerie Lotus-Notes ™,...

  Les demandes d'accès lui parviennent sous trois formes :
  - oralement, dans ce cas une confirmation écrite est demandée
  - via *Mémo formaté*
  - via une application développée en interne décrite dans le chapitre «5.2.4 'Point de vue d'un développeur d'une application de gestion d'accès' »

  Le traitement des demandes est fortement délocalisé grâce aux *coordinateurs décentralisés*. Leur rôle est de prendre en charge les demandes concernant des ressources précises ou celles concernant des groupes d'utilisateurs déterminés. L'acceptation d'une demande passe toujours par l'accord du propriétaire de la ressource qui délègue généralement ce pouvoir à un *coordinateur décentralisé*.
  Dans cette organisation, les coordinateurs décentralisés s'occupent de toutes les demandes standardisées et récurrentes et assurent la communication entre les intervenants, quant au gestionnaire d'accès, il prend en charge toutes les demandes qui concernent plus globalement l'entreprise ou qui sont très spécifiques ou spéciales.

  Il existe en fait deux applications pour effectuer cette gestion. La première application gère l'ensemble des accès mainframes et la seconde les accès aux *ressources NT*, *Netware*, *Vax* et *RAS*.

  La liste des utilisateurs est importée quotidiennement dans le système à partir d'informations provenant du service *GRH* en charge des ressources humaines de la société. Aucun utilisateur ne peut être créé directement dans le système de gestion d'accès.

  Généralement la gestion des accès se fait sur une base individuelle. Des demandes qui concernent plusieurs personnes doivent être faites séparément. Les gestionnaires d'accès souhaitent que les demandes groupées restent exceptionnelles.

  Points forts du système existant

  - Procédure formelle pour le traitement des demandes.
  - Distribution du traitement des demandes (décentralisation).

  Points faibles du système existant

  - Plusieurs applications pour traiter la problématique des accès (interfaces multiples, complexité due au nombre, sources d'erreurs, ...).
  - Pas d'historique des actions entreprises.
  - Archivage non uniformisé.
  - Formulaires de demande organisés par plate-forme système ; nécessité pour le demandeur de connaître les caractéristiques techniques de sa demande (manque de convivialité et de simplicité).
  - Manque d'uniformisation et d'intégration des différents moyens d'introduire une demande d'accès.

- Guy est responsable de la gestion des emplacements de parking dans les zones piétonnes du centre-ville namurois.

  Il existe dans la commune de Namur quatre zones piétonnes dont l'accès est protégé par une dizaine de bornes amovibles. Ces bornes sont autonomes, ce qui signifie que leur configuration est locale. L'accès à ces zones piétonnes nécessite l'usage d'une carte magnétique qui doit être présentée à l'entrée de ces zones. L'accès à ces zones est réglementé par des directives émises par la ville de Namur. Par exemple, les services de secours bénéficient d'un accès permanent tandis que les fournisseurs disposent d'un accès uniquement pendant les heures de livraison. Les riverains, quant à eux, peuvent demander un accès uniquement s'ils ne disposent pas d'un garage ou d'un emplacement de parking réservé dans un rayon d'un kilomètre.
  Deux services distincts interviennent dans la gestion de ces accès. Le service stationnement de la police de Namur effectue les vérifications d'usage (les contrôles de légitimité), active la carte magnétique et communique les informations au service de la régie urbaine de l'équipement de la Ville de Namur qui est chargé de configurer les bornes amovibles de la zone spécifiée, ce qui implique l'encodage du numéro de la carte magnétique ainsi que les heures d'accès pour chaque borne impactée. La demande est introduite uniquement auprès du service stationnement de la police de Namur qui préviendra le demandeur lorsque tout est en ordre.

  Le traitement de ces demandes est entièrement manuel, ce qui ne pose pas de problème à l'heure actuelle vu le faible nombre de demandes. Toutefois, un système centralisé de configuration de ces bornes est à l'étude.

  Points forts du système existant

  - Vérification stricte établie par la Police suivant une procédure formelle

  Points faibles du système existant

  - Traitement manuel des demandes (aucune automatisation)
  - Pas de possibilité d'interrogation des accès existants
  - Désynchronisation (configuration) possible des bornes amovibles d'une même zone d'accès.

5.2.3 Point de vue des gestionnaires de ressources

- Christian gère un ensemble de composants réseaux dans le cadre d'un extranet privé reliant notamment de nombreuses communes belges à une société bancaire.

  Les demandes d'intervention lui parviennent par téléphone, courrier interne ou courrier électronique.
  Après la réception de la demande d'intervention, il étudie la faisabilité de celle-ci et informe les gestionnaires des composants impliqués qui ne sont pas sous sa responsabilité. Ensuite, il effectue l'intervention proprement dite. Enfin, il avertit le demandeur pour qu'il entreprenne des tests de fonctionnement.

  Il utilise l'application *Info Management* pour archiver les interventions effectuées. Cette application lui procure un historique de ce qui a été entrepris pour chaque demande.

  Points forts du système existant

  - Historique succinct des interventions via Info Management.

  Points faibles du système existant

  - Pas de procédure formelle pour les demandes d'interventions.
  - Difficulté de synchronisation entre les différents intervenants.
  - Communication directe de chaque gestionnaire de ressources avec le bénéficiaire, ce qui est une source de confusion et renforce le sentiment de complexité.

- Emmanuel est administrateur système de plus de 30 serveurs de bases de données Sybase et Oracle sur des plate-formes Unix et NT. Son rôle est de veiller au bon fonctionnement de ceux-ci et de gérer la définition des utilisateurs.

  Il ne gère pas personnellement les demandes d'accès sur les bases de données, cette gestion est laissée aux « Database Administrators ». Par contre, il assure la définition des utilisateurs, c'est à dire la création des logins et des mots de passe initiaux. Actuellement, il existe environ 300 logins définis sur ces systèmes.

  Les demandes de création de compte utilisateurs sont envoyées par les responsables des applications sous la forme de *Record Infoman*. Le contrôle de légitimité est effectué par ces derniers, Emmanuel ne vérifie que l'identité du demandeur.
  Il extrait manuellement les informations du *Record Infoman* pour alimenter un script qui génère automatiquement les logins. Il accepte également les demandes par téléphone des utilisateurs finaux pour la réinitialisation de leur mot de passe. Il n'existe actuellement pas de procédure pour la suppression de ces logins. Le feedback est assuré par la clôture du *Record Infoman* qui signifie que tout est en ordre. Emmanuel ne dispose pas d'outil lui permettant d'effectuer des requêtes sur les définitions déjà établies.

  Points forts du système existant

  - Procédure formelle pour la création des logins
  - Feedback systématique (mais passif)

  Points faibles du système existant

  - Manipulation fastidieuse pour l'extraction des informations du *Record Infoman*
  - Pas de suppression des comptes utilisateur inutilisés
  - Pas de possibilité de requêtes

Remarque : nous n'avons pas recueilli d'autres avis pour cette catégorie d'acteur car au moment de cette enquête, cette fonction faisait partie de notre mission à tous les deux.

5.2.4 Point de vue d'un développeur d'une application de gestion d'accès

Remarque préliminaire : contrairement à ce qui précède, les propos recueillis ici ne sont pas issus de l'enquête mais plutôt d'une interview personnalisée.

- Claude a développé une application prenant en charge la problématique de la gestion des demandes d'accès dans une banque. Actuellement, les ressources concernées par cette application comprennent la plate-forme NT, Netware, Vax et l'accès distant RAS) . Bientôt, le traitement des demandes d'accès à Internet sera également opérationnel. L'application est basée sur le workflow proposé par Lotus-Notes©. Le système comprend actuellement environ 10.000 utilisateurs et 100.000 accès existants.

  La liste des acteurs concernés par cette application est figée. Elle comprend l'utilisateur qui effectue la demande, le cadre de direction dont dépend le demandeur qui représente l'autorité d'approbation et qui valide ou pas la demande, le coordinateur décentralisé de sécurité qui gère les accès des utilisateurs pour une zone géographique donnée et enfin le gestionnaire de ressources.

  Les modifications sur les accès établis ne sont pas possibles sans passer à nouveau dans un workflow. Toutefois, suite à la demande de certains gestionnaires de ressources, il leur est possible d'accéder directement à la base de données du système sans passer par un nouveau workflow.

  A l'origine, seul l'utilisateur pouvait effectuer une demande de suppression d'un de ses accès Suite à la demande des *coordinateurs décentralisés*, ces derniers peuvent également supprimer d'initiative un accès d'un utilisateur sous leur responsabilité. Le système possède également un agent qui, lors de l'importation de la liste des utilisateurs provenant de la gestion des ressources humaines, supprime les accès des utilisateurs non repris dans cette liste.

  Il existe diverses possibilités de reporting avec plusieurs niveaux de sécurité d'utilisation. Ce reporting est développé à la demande et donc il n'est pas possible d'effectuer des requêtes personnelles. Le résultat de ces requêtes est imprimable.

  La reprise des accès existants est exécutée grâce à l'importation de fichiers textes fournis par les gestionnaires de ressources. Notons que les gestionnaires de ressources ont la possibilité technique dans la majorité des cas de créer ou de modifier des accès existants. Ceci constitue alors une source importante de discordance entre les accès réels présents sur les ressources et l'inventaire des accès théoriques présent dans l'application.

  La communication entre le système et les gestionnaires de ressources utilise le courrier électronique comme vecteur. Cela implique que les gestionnaires, après avoir réceptionné la demande, doivent effectuer une intervention manuelle pour configurer le composant. Certains gestionnaires de ressources ont émis le souhait d'automatiser cette tâche.

  Tous les acteurs participant au traitement d'une demande peuvent s'enquérir du statut de celle-ci en interrogeant le système.

Points forts du système existant

- Application disponible rapidement et qui satisfait pleinement les responsables.
- Application parfaitement intégrée dans Lotus-Notes qui est le standard actuel de groupware de l'entreprise.
- Les acteurs peuvent consulter le déroulement de la demande.
- Introduction d'une notion de « service » qui cache une partie de la complexité au demandeur

Points faibles du système existant

- Application non portable.
- Application monolithique où tout changement des fonctionnalités nécessite un bouleversement de l'application toute entière.
- Scénarios figés.
- Pas de gestion des groupes.
- Possibilité limitée de suppression des accès.
- Impossibilité de créer des requêtes personnelles. Claude doit les développer lui-même.
- Possibilité limitée pour les gestionnaires de ressources d'interagir avec le système.
- Prolifération des applications de Workflow.

5.3 *Liste des acteurs*

Sur la base des interviews effectuées sur la gestion des accès dans trois domaines d'applications différents, nous pouvons déduire à présent la liste des acteurs principaux liés à notre système :

- *L'initiateur de la demande que l'on peut appeler aussi « le demandeur »*
  Il s'agit de la personne qui entreprend la démarche de requérir un nouvel accès ou la modification d'un accès existant.

- *Le bénéficiaire de la demande*
  Il s'agit de la personne qui va utiliser l'accès demandé. Le bénéficiaire d'une demande peut être l'initiateur lui-même ou une autre personne. Il peut représenter aussi un groupe de personnes.

- *L'autorité d'approbation*
  Il s'agit de la ou des personnes chargée(s) d'approuver ou non la demande. Il est également possible que cette approbation ne soit pas effectuée par des personnes mais par un ensemble de règles qui sont appliquées par le système pour vérifier la légitimité de la demande. Ces règles auront été établies par les personnes représentant alors l'autorité d'approbation.

- *Le gestionnaire d'accès*
  Cette personne est responsable de la gestion globale des accès et de la sécurité découlant de ceux-ci.

- *L'administrateur système*
  Il s'agit de la personne qui prend en charge l'administration du système de gestion des accès. En général, c'est le gestionnaire d'accès qui remplit cette fonction.

- *Le(s) gestionnaire(s) de(s) ressource(s)*
  Les gestionnaires de ressources représentent les personnes chargées de veiller au bon fonctionnement des ressources dont elles ont la responsabilité.

- *Le propriétaire de ressource*
  Il s'agit de la personne qui possède la ressource et qui assume l'intégrité de celle-ci.

- *L'utilisateur du système*
  Toute personne interagissant avec le système est considérée comme utilisateur du système.

5.4 Analyse des besoins (synthèse des interviews)

Nous attirons l'attention du lecteur sur le fait que les points abordés dans ce chapitre ne constituent pas les fonctionnalités du système proposé mais uniquement une synthèse des besoins émis par les participants à l'enquête. De ce fait, certains points décrits ci-dessous sont d'application uniquement pour un domaine particulier.

5.4.1 Point de vue des demandeurs et des utilisateurs bénéficiaires

- *Simplicité des formulaires*
  Le demandeur souhaite formuler sa demande de manière unique et uniforme. Le formulaire de demande devra être accessible même pour une personne sans notions techniques, c'est à dire que les services proposés sur celui-ci ainsi que les renseignements demandés devront être présentés avec des termes compréhensibles pour le demandeur. Le formulaire de demande doit cacher au demandeur la complexité de la demande.

- *Convivialité*
  Les utilisateurs du système de gestion des demandes d'accès souhaiteraient que celui-ci soit agrémenté de fonctionnalités améliorant la convivialité ; comme par exemple : une aide en ligne, des coordonnées de personnes de contact, des messages informatifs, messages d'avertissement avant expiration d'un accès.

- *Retour d'informations*
  Dès la prise en charge de la demande par le système, celui-ci devra prévoir des retours d'informations réguliers et systématiques à destination des intéressés, et ceci jusqu'à la fin de son traitement. Ces messages sont donc envoyés « spontanément » par le système à TOUS les acteurs concernés.

- *Suivi de la demande*
  Le demandeur et le bénéficiaire souhaitent être avertis de tout événement important découlant de la demande d'accès. Par exemple, la disponibilité de ce nouvel accès devra leur être notifiée. De plus, ils doivent également pouvoir suivre l'état d'avancement de toute demande les concernant.

- *Interrogation du système*
  Le bénéficiaire souhaite pouvoir effectuer certaines requêtes comme, par exemple, consulter la liste de ses accès existants ou la liste des ressources auxquelles il lui est possible de demander accès.

- *Reprise de l'existant*
  Si un nouveau système de gestion doit apparaître, le bénéficiaire ne souhaite pas devoir refaire des demandes pour tous ses accès existants. Une procédure de reprise de l'existant doit être prévu.

- *Disponibilité et rapidité*
  Le système de demande d'accès doit être disponible à tout moment (jour et nuit) car, dans la majorité des cas, les accès sont demandés lorsque le besoin est déjà présent. Le demandeur doit pouvoir introduire sa demande de tout endroit où il peut se trouver dans l'exercice de ses fonctions. Pour les mêmes raisons que ci-dessus, la demande d'accès doit être traitée rapidement et même parfois immédiatement.

- *Désactivation des accès inutiles*
  Certains utilisateurs demandent à avoir la possibilité de demander la suppression de certains de leurs accès une fois ceux-ci jugés inutiles. Il est à noter que dans les domaines où une tarification est liée à un accès, cette fonctionnalité est vivement souhaitée.

5.4.2 Point de vue des gestionnaires d'accès

- *Contrôle de faisabilité*
  Le système proposé doit se prononcer le plus tôt possible sur le caractère réaliste de la demande. La faisabilité doit également être vérifiée avec chaque gestionnaire de ressource impactée. Dans certains cas, le gestionnaire d'accès peut refuser la demande et proposer une solution alternative.

- *Archivage des demandes d'accès*
  Le système doit garder un historique de toutes les demandes d'accès, notamment dans le but de pouvoir déterminer les accès d'un utilisateur et les raisons qui ont justifié leur accord.

- *Activation retardée des accès*
  Le système offrira la possibilité d'activer un accès dans le futur (définition anticipée).

- *Gestion centralisée*
  En général, ces personnes souhaitent la centralisation de la gestion des accès en permettant quand même de déléguer certaines tâches à des *coordinateurs décentralisés*. Le gestionnaire d'accès souhaite que toutes les demandes d'accès soient traitées dans son service ou via son système.

- *Simplification et économie*
  L'utilisation d'un système de gestion des accès devrait permettre en premier lieu la simplification de la gestion, la diminution des coûts ainsi que des erreurs de traitement.

- *Reporting, auditing et alarming*
  Selon eux, le système devrait fournir des possibilités avancées de *reporting*, *auditing* et *alarming*. Le reporting consiste à effectuer des interrogations sur la base d'informations du système. L'auditing signifie que toutes les actions entreprises par le système ou par les acteurs sont enregistrées. Enfin, l'alarming signifie que certains événements générés par le système ou des actions entreprises par les acteurs se traduiront par un message d'avertissement à destination de l'administrateur du système.

- *Règles de bonne conduite*
  Toute action entreprise par le système de gestion des accès doit être conforme à la politique de sécurité de l'organisation (*Security policies*) et à ses contraintes organisationnelles.

- *Suivi de l'état d'avancement*
  Le suivi du déroulement d'une demande d'accès est primordial. Le gestionnaire désire pouvoir consulter l'état d'une demande, les blocages éventuels et déterminer pourquoi une demande a été refusée.

- *Contrôle des inconsistances*
  Dans certains cas, un contrôle croisé est nécessaire pour détecter les incohérences entre la situation théorique (déterminée par le système) et la réalité (les accès réels aux ressources).

- *Visibilité limitée*
  Dans certains cas, le gestionnaire des accès souhaite pouvoir restreindre la liste des services proposés au demandeur.

- *Organisation fonctionnelle*
  Le système doit tenir compte de l'aspect fonctionnel de l'organisation, c'est à dire qu'il devrait pouvoir refléter la structure organisationnelle de celle-ci. Il s'agit notamment d'intégrer les bénéficiaires, les ressources et les accès déjà opérationnels.

- *Gestion des évènements*
  Le système devrait pouvoir réagir à une série d'évènements tels que, par exemple, la mutation ou la suppression d'un bénéficiaire, la mise hors service d'une ressource. Il devra donc générer automatiquement les demandes d'interventions pour adapter les accès à la nouvelle situation.

- *Demande de suppressions*
  La plupart des systèmes analysés ne permettent pas à un bénéficiaire de demander la suppression de certains de ses accès. Le système proposé devrait offrir cette possibilité. La suppression automatique des accès inutiles serait aussi un plus indéniable.

- *Historique et archivage*
  Il est également demandé d'archiver de manière centralisée et uniforme les demandes d'accès et leurs justifications. Lorsqu'un workflow de traitement de la demande existe, un archivage a souvent lieu mais il est rarement uniformisé.

- *Interrogation du système*
  Le gestionnaire souhaite également pouvoir consulter les informations suivantes :
  - la liste des accès d'un utilisateur
  - la liste des accès par service
  - la liste des accès par groupe

5.4.3 Point de vue des gestionnaires des ressources

- *Uniformité des demandes d'intervention*
  Le gestionnaire de ressources désire recevoir les ordres de modifications des accès à la ressource de manière uniforme.

- *Automatisation des tâches d'intervention*
  Certains gestionnaires de ressources souhaiteraient automatiser les interventions sur des composants dont ils ont la gestion.

- *Accessibilité*
  L'accessibilité de l'interface de gestion du système est importante. Le gestionnaire de ressources doit pouvoir accéder au système de tout endroit où il peut se trouver dans l'exercice de ses fonctions.

- *Auditing*
  Le gestionnaire de ressource souhaite que le système lui offre la possibilité d'effectuer des requêtes multicritères sur l'historique des interventions effectuées. Le résultat de ces requêtes doit pouvoir être imprimé. Il souhaite bénéficier d'un historique des interventions effectuées sur une ressource.

- *Interface du système*
  L'interface d'utilisation du système doit être simple à utiliser, rapide, conviviale.

- *Tests*
  La mise en œuvre d'un accès nécessite souvent un ensemble de tests afin de vérifier le bon fonctionnement de celui-ci. Ceci doit donc être prévu dans le système.

5.5 *Observations*

En discutant avec les différentes personnes que nous avons interrogées, nous avons non seulement pu recueillir leur point de vue mais avons aussi eu l'occasion de comparer les situations de chacun et d'avoir ainsi une vue transversale de la problématique. Nous avons observé quelques points intéressants que nous exposons ici.

- *Divergence de points de vue*
  Le point de vue des gestionnaires d'accès et celui des utilisateurs sont souvent en contradiction. Les intérêts des uns sont parfois incompatibles avec ceux des autres. Par exemple, le souci majeur d'un gestionnaire d'accès (ou de sécurité) est d'octroyer le minimum d'accès à un utilisateur qui lui sont nécessaires pour réaliser ses tâches ; celui d'un utilisateur est de se ménager le moins d'obstacles possibles dans le cadre de la réalisation de ses tâches. Le fait de devoir demander ou se voir refuser des accès pour une question de politique globale de sécurité est une source fréquente de discordance.

- *Prise en charge de la coordination des intervenants et donc des interventions*
  Les gestionnaires de ressources sont souvent isolés des utilisateurs; ce sont les gestionnaires d'accès qui assurent la coordination des opérations relatives aux demandes d'accès.

- *Problème de communication*
  Notre expérience et les observations qui résultent de l'enquête, nous laisse entrevoir un problème important de communication entre les bénéficiaires/demandeurs et les gestionnaires d'accès/de ressources. Les premiers s'expriment en termes fonctionnels et les seconds en termes techniques. Un système qui offrirait la possibilité de « traduire » les termes des uns en ceux des autres présenterait une plus value indéniable.

- *Refus ou impossibilité technique de réalisation de la demande à posteriori*
  Dans certains cas, la demande est approuvée, mais à un certain moment du traitement de celle-ci, un refus ou une impossibilité de réaliser l'intervention apparaît. Ce problème, heureusement pas trop fréquent, provoque un certain inconfort pour le demandeur. Il n'existe, selon nous, pas de moyen pour éviter totalement le problème mais bien un certain nombre de mesures qui permettent de le minimiser (exemple : bonne tenue de l'inventaire des ressources accessibles, bonne communication bi-directionnelle entre les gestionnaires d'accès et les gestionnaires de ressources, ...)
  De plus, si une demande concerne plusieurs ressources, il se peut que certains accès soient déjà établis sur certaines de ces ressources avant que l'impossibilité ne soit annoncée. Ce qui implique l'obligation de disposer d'un moyen d'annuler toutes les actions déjà entreprises pour cet accès.

6 Liste des exigences

6.1 *Introduction*

Dans ce chapitre, nous présentons dans un premier temps la liste des exigences exprimées lors de l'enquête, des interviews et de notre analyse de celles-ci en les regroupant en fonction de leur origine. Ensuite nous synthétisons ces exigences d'un point de vue fonctionnel en les classifiant selon trois catégories :

- les exigences fonctionnelles qui seront couvertes par SAGA
- les exigences fonctionnelles qui ne seront pas couvertes par SAGA
- les contraintes non fonctionnelles

Lors de cette étape, les objectifs du système – que nous avons vu au chapitre 3.1 Objectifs de SAGA – ont également été convertis en contraintes non fonctionnelles.

Chaque exigence est précédée d'un numéro d'ordre qui servira à la traçabilité au niveau de l'étape de synthèse.

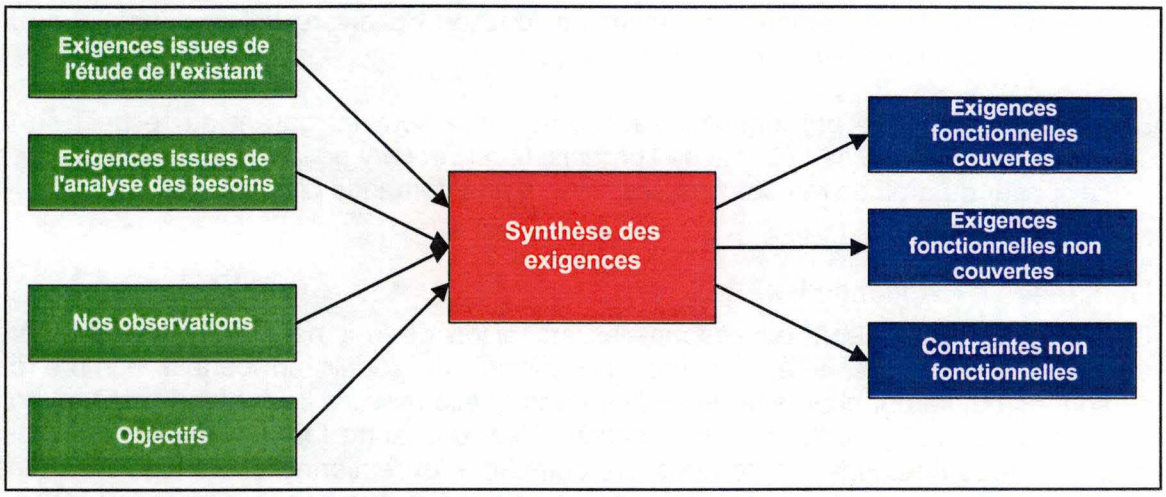

Figure 2 - Synthèse des exigences

6.2 *Exigences issues de l'étude de l'existant*

1) Contrôles de la légitimité de la demande

La validité de toute demande devra être vérifiée par le système.
Celle-ci devrait être contrôlée en fonction de trois critères :

- Légitimité : le système doit vérifier si le demandeur est autorisé à effectuer une demande pour un bénéficiaire et un service donné en fonction de règles déterminées. Ce contrôle peut se baser sur des informations fournies par le demandeur ou par un système extérieur. Dans certains cas, la légitimité de la demande est subordonnée à l'existence d'un contrat (d'engagement d'une personne, de télémaintenance, d'un abonnement, ...). Le lien entre le demandeur et les personnes visées par le contrat devra également être vérifié.

- Approbation : Une autorité d'approbation doit donner son accord pour l'utilisation du service demandé par le bénéficiaire. L'autorité d'approbation peut comprendre entre autres : le propriétaire des ressources impactées, l'autorité hiérarchique du bénéficiaire et/ou du demandeur,...). La composition de cette autorité d'approbation peut varier d'une demande à l'autre en fonction de divers critères (nature de la demande, service demandé, bénéficiaire, ...).

- Droit de veto : Le gestionnaire d'accès se porte souvent garant du respect de la politique générale de sécurité de l'organisation (*Security policies*). A ce titre, il devra disposer du droit de refuser souverainement une demande d'accès.

2) Demandes exceptionnelles

Certaines demandes sont exceptionnelles en raison de leur nature ou leur caractère unique. Dans ce cas, il est souvent nécessaire de réunir un certain nombre de personnes qui seront chargées de vérifier dans quelle mesure la demande met ou non en péril la sécurité globale de l'organisation. C'est une autre forme de validation de la légitimité de la demande. Selon nous, les demandes exceptionnelles concernent celles qui ne rentrent pas dans le canevas habituel, c'est à dire que soit, le demandeur ne voit pas dans sa liste le service souhaité, soit ce service n'existe pas encore. Dans tous les cas, le système devrait offrir la possibilité de traiter ce type de demande.

3) Délégation

Dans certains domaines, qui sont particulièrement spécialisés ou pour lesquels les ressources et les utilisateurs sont géographiquement dispersés, la gestion des accès (et donc des demandes d'accès) est déléguée à des personnes que l'on peut appeler *coordinateurs décentralisés* . Ces personnes ont pour mission de diminuer la charge de travail du gestionnaire d'accès en gérant les accès pour un ensemble de ressources déterminées ou pour un ensemble de personnes donné. Le système devra autoriser cette délégation des tâches en limitant toutefois le champ d'action.

4) Accès de base

Le système devra pouvoir gérer les accès de base d'une personne, c'est à dire les accès qui lui sont nécessaires pour exercer sa fonction primaire. Il s'agit donc de lui accorder des accès par défaut.

5) Demande à effet immédiat

En cas d'urgence, un utilisateur peut demander un accès immédiat à certaines ressources déterminées, à condition qu'il le justifie. Dans ce cas, il n'y a pas de contrôle humain et l'intervention est réalisée directement par le système sur la ressource en question. Il s'agit d'une liste d'accès restreinte. Ces accès ont souvent une période de validité très courte. Le système devrait prendre en charge ce genre de demande immédiatement.

6) Demande motivée

Dans certains cas, le gestionnaire d'accès exige du demandeur qu'il justifie sa demande. Le système devrait permettre à ce dernier d'introduire un texte de motivation ou d'établir le lien avec un document existant.

7) Simplification et économie

L'utilisation d'un système de gestion des accès devrait permettre en premier lieu la simplification de la gestion, la diminution des coûts ainsi que des erreurs de traitement.

8) Disponibilité et accessibilité

Le système de demande d'accès doit être disponible à tout moment (jour et nuit) car, dans la majorité des cas, les accès sont demandés lorsque le besoin est déjà présent. Chaque utilisateur du système doit pouvoir également accéder au système de tout endroit où il peut se trouver dans l'exercice de ses fonctions.

6.3 Exigences issues de l'analyse des besoins

9) Simplicité des formulaires

Le formulaire de demande doit cacher au demandeur la complexité de la demande. Celui-ci devrait donc être accessible même pour une personne sans notions techniques, c'est à dire que les services proposés sur celui-ci ainsi que les renseignements demandés devront être présentés avec des termes compréhensibles pour le demandeur.

10) Convivialité

Les utilisateurs du système souhaiteraient que celui-ci soit agrémenté de fonctionnalités améliorant la convivialité, comme : l'aide en ligne, les coordonnées de personnes de contact, des messages informatifs ainsi que les messages d'avertissement avant l'expiration d'un accès.

11) Retour d'informations et suivi de la demande

Dès la prise en charge de la demande par le système, celui-ci devrait informer régulièrement et systématiquement les intéressés de l'état d'avancement du traitement de leur demande. Ces messages devraient être envoyés « spontanément » par le système à tous les acteurs concernés. Une estimation du délai de mise à disposition de l'accès demandé devrait également être fournie au demandeur.
De plus, les personnes intéressées doivent également pouvoir suivre l'état d'avancement de toute demande les concernant en interrogant le système.

12) Interrogation du système

Les différents acteurs souhaitent pouvoir interroger le système afin de connaître par exemple :

- Liste des accès par bénéficiaire, par service
- Liste des personnes, des services et des ressources
- Liste des chemins d'accès (CAs) transitant par une ressource donnée
- Liste des services utilisant une ressource donnée
- Liste des demandes en cours, non satisfaites
- Liste des demandes d'interventions par gestionnaire de ressources
- Estimation du délai moyen du traitement d'une demande
- Étude d'impact sur les accès en cas de modification ou déplacement d'une ressource

Ce type de requête pourrait utiliser des critères de recherche multiples. Le résultat de celles-ci devra pouvoir être imprimé.

13) Reprise de l'existant

Lors de la mise en place du système, celui-ci devrait pouvoir reprendre intégralement la situation existante qui comprend notamment les accès, les ressources et les personnes.

14) Tests de fonctionnement de l'accès

Les gestionnaires de ressources sont intéressés par la possibilité de tester, une fois la demande réalisée, le bon fonctionnement de l'accès. Le système devrait proposer une batterie de tests en relation avec le service proposé.

15) Contrôle de faisabilité

Le système proposé devrait se prononcer au plus tôt sur le caractère réaliste de la demande. La faisabilité doit également être vérifiée avec chaque gestionnaire de ressource impactée. Dans certains cas, le gestionnaire d'accès peut proposer une solution alternative à la demande formulée.

16) Activation retardée des accès

Le système devrait offrir la possibilité d'activer un accès dans le futur (définition anticipée).

17) Automatisation des tâches d'intervention

Certains gestionnaires de ressources souhaiteraient automatiser les interventions sur des ressources dont ils ont la gestion, c'est à dire que dans ce cas, le système configure directement les ressources impactées.

18) Conformité aux règles de l'organisation

Toute action entreprise par le système de gestion des accès devrait être conforme à la politique de sécurité de l'organisation (*Security policies*) et à ses contraintes organisationnelles.

19) Reporting

Le système proposé devrait pouvoir générer et imprimer des rapports d'activité tels que des statistiques diverses ou une liste des interventions en cours.

20) Journalisation des événements

La journalisation signifie que tous les évènements et actions entreprises par le système ou les différents acteurs sont enregistrés. Le système devrait comporter ce type de fonctionnalité.

21) Gestion des alarmes

La gestion des alarmes signifie que le système, suite à certains événements, envoie un message d'avertissement. Le système devrait comporter ce type de fonctionnalité.

22) Visibilité limitée

Dans certains cas, le gestionnaire des accès souhaite pouvoir restreindre la liste des services proposés au demandeur.

23) Organisation fonctionnelle

Le système doit tenir compte de l'aspect fonctionnel de l'organisation, c'est à dire qu'il devrait pouvoir refléter la structure organisationnelle de celle-ci. Il s'agit notamment d'intégrer les bénéficiaires, les ressources et les accès déjà opérationnels.

24) Gestion des évènements

Le système devrait pouvoir réagir à une série d'évènements tels que, par exemple, la mutation ou la suppression d'un bénéficiaire, la mise hors service d'une ressource. Il devrait donc générer automatiquement les demandes d'interventions pour adapter les accès à la nouvelle situation.

25) Uniformité des demandes d'intervention

Le gestionnaire de ressources désire recevoir les ordres de modifications de manière uniformisée.

26) Historique et archivage

Le système devrait permettre de reconstituer l'historique des accès d'un utilisateur ainsi que les raisons qui ont motivé l'octroi de ceux-ci. Ceci comprend notamment l'archivage des actions entreprises par le système pour mettre en œuvre la demande et l'archivage de toutes les demandes d'accès et des pièces justificatives.

27) Gestion centralisée

Le gestionnaire d'accès souhaite que toutes les demandes d'accès soient traitées par un système unique.

28) Traitement rapide de demandes d'accès

Le bénéficiaire souhaite que le délai de traitement de sa demande soit réduit à un minimum.

29) Suppression des accès

Certains utilisateurs demandent à avoir la possibilité de demander la suppression de certains de leurs accès une fois ceux-ci jugés inutiles. Il est à noter que dans les domaines où une tarification est liée à un accès, cette fonctionnalité est vivement souhaitée. Le système proposé devrait offrir cette possibilité.

6.4 Exigences déduites de nos observations

30) Authentification des utilisateurs du système

Chaque utilisateur du système devrait être authentifié afin de déterminer son profil d'utilisation, c'est à dire l'ensemble des actions qu'il peut entreprendre dans le système.

31) Traitement des accès inutiles ou inutilisés

L'utilisation d'un système qui réagit aux événements pour adapter les accès (principalement les suppressions), ne garantit pas que tous les événements qui peuvent avoir un impact sur l'inventaire des accès lui soient communiqués, il subsiste donc toujours le risque que des accès soient inutilisés ou inutiles. Afin de minimiser ce risque, le système devrait effectuer une vérification périodique de la nécessité des accès existants.

32) Inventaire des accès existants

On peut dire que même si le système permettait de réaliser un inventaire des demandes d'accès, ce n'est pas pour cela que l'on pourrait en déduire de manière sûre l'inventaire des accès effectifs. En effet, on ne peut malheureusement pas contraindre les gestionnaires de ressources à gérer tous les accès à leurs ressources via le système de gestion centralisée. On peut à tout le moins l'imposer « politiquement ».
Pour minimiser ces écarts entre la réalité et l'inventaire, le système devrait effectuer des contrôles croisés permettant de détecter les inconsistances. Une synchronisation régulière serait également un bon palliatif à cette problématique.

33) Gestion des utilisateurs et des groupes

Le mode de création des utilisateurs dans le système dépend du domaine dans lequel on se trouve et probablement du public visé. En effet, dans une entreprise on peut imaginer aisément qu'il est en relation étroite avec la gestion des ressources humaines. Mais dans le domaine des parkings, il s'agit probablement d'une gestion « à la demande ». Le système devrait pouvoir externaliser la gestion des utilisateurs et des groupes ou se synchroniser avec un système existant (par exemple par un mécanisme d'importation) mais également pouvoir dans certains cas assurer lui-même cette gestion.

34) Détermination automatique des ressources impactées par la demande

Dès la prise en charge de la demande, le système devrait pouvoir déterminer la liste des ressources sur lesquelles il faudra intervenir.

35) Tâches clé et système centralisé

Chaque acteur doit pouvoir effectuer ses tâches clé de gestion des demandes d'accès et de gestion des accès via le système proposé. Par exemple, l'administrateur du système doit avoir la possibilité de gérer tous les objets de base tels que les personnes, les groupes de personnes, les profils et les services.

36) Intégration dans l'environnement informatique existant

Le système SAGA devrait s'intégrer avec les systèmes et applications informatiques existantes dans l'organisation.

37) Exportation d'information

Afin d'améliorer également l'intégration avec la situation existante, le système doit pouvoir exporter certaines de ses informations.

6.5 Synthèse des exigences du système

Le moment est venu de déterminer avec précision ce qui sera pris en charge par notre système et ce qui ne le sera pas. C'est l'occasion de fixer donc les limites du système.
Le tableau suivant reprend la synthèse des exigences que devrait offrir notre système. La première colonne contient un numéro de séquence qui permettra de vérifier lors de l'analyse si notre système couvre toutes ces exigences. La seconde colonne donne une brève description de l'exigence.

6.5.1 Exigences fonctionnelles couvertes par notre système

1	**Gestion des utilisateurs et des groupes** Le système devrait proposer différents modes d'acquisition des utilisateurs et des groupes, notamment grâce à des formulaires, par importation de données ou grâce un lien avec une application externe.
2	**Traitement et suivi automatique des demandes.** Dés leur introduction, les demandes sont prises en charge par le système. Les différents acteurs devraient pouvoir consulter à tout moment le statut d'une demande les concernant et être avertis en cas de modification significative du statut de celle-ci.
3	**Distribution des interventions.** Les informations nécessaires à la configuration des ressources pour la mise en œuvre d'un accès à un service devraient être transmises de manière uniformisée à chaque gestionnaire de ressources.
4	**Administration étendue des objets** Le système devrait offrir à l'administrateur du système une panoplie d'outils lui permettant de gérer aisément (introduire et maintenir à jour) les différents objets du système d'information, notamment : o Les ressources o Les services o Les profils o Les personnes et groupes de personnes
5	**Adaptation automatique des accès suite à des événements.** Lors de la modification par exemple, d'un groupe d'utilisateur, d'un profil ou d'un service, le système adapte de manière automatique les accès impactés. Ainsi, lors de la suppression d'un utilisateur, des demandes d'intervention sont générées afin de désactiver tous ses accès.
6	**Suppression des accès** L'utilisateur devrait pouvoir supprimer ses accès personnels.

7	**Contrôles sur la demande d'accès** La validité d'une demande devrait être contrôlée par : o la vérification par le système en fonction de règles si le demandeur est autorisé à effectuer une demande pour un utilisateur et un service donné. (légitimité de la demande). Ce contrôle peut se baser sur des informations fournies par le demandeur ou par un système extérieur. o la vérification par l'autorité d'approbation si l'utilisateur peut bénéficier du service demandé. o La vérification de la motivation de la demande. o La vérification optionnelle par le gestionnaire d'accès de la validité de cette demande.
8	**Interrogation du système et génération de rapport** Le système devrait permettre d'effectuer des requêtes diverses et d'imprimer les résultats. On peut citer notamment : o Liste des accès par bénéficiaire, par service o Liste des personnes, des services et des ressources o Liste des chemins d'accès (CA) transitant par une ressource donnée o Liste des services utilisant une ressource données o Liste des demandes en cours, non satisfaites o Liste des demandes d'interventions par gestionnaire de ressources o Estimation du délai moyen du traitement d'une demande o Étude d'impact sur les accès en cas de modification ou déplacement d'une ressource
9	**Possibilité de déléguer la gestion du système.** Le gestionnaire d'accès devrait pouvoir déléguer certaines tâches de gestion du système à d'autres personnes (gestionnaires décentralisés).
10	**Gestion des accès de base** Lors de l'ajout d'un utilisateur, le système devrait générer si nécessaire les demandes d'intervention qui correspondent à ses accès de base.
11	**Vérification de la validité des accès** Le système devrait vérifier si les accès sont encore utiles et utilisés. Il devrait prendre les actions nécessaires pour désactiver les accès inutiles ou arrivés à échéance.
12	**Reprise de l'existant** Lors de la mise en place du nouveau système, il devrait exister des procédures permettant d'importer des informations provenant des systèmes existants.
13	**Activation différée** Il devrait être possible de choisir la date d'activation d'un accès.
14	**Possibilité d'exporter les informations du système** Le système devrait pouvoir exporter sous divers formats les informations dont il assure la gestion.

15	**Sécurité** La sécurité du système devrait être assurée, notamment par : o Une authentification préalable de tous les utilisateurs de ce système o La possibilité de définir les actions que chacun peut entreprendre ou pas o La possibilité de définir les informations que chacun peut accéder
16	**Automatisation de la recherche des ressources impactées** Dès la prise en charge de la demande, le système devrait pouvoir déterminer la liste des ressources sur lesquelles il faudra intervenir.
17	**Archivage** Le système devrait enregistrer toutes les informations nécessaires à la reconstitution de l'historique des actions entreprise par l'ensemble des acteurs (y compris le système lui-même)
18	**Contrôle de faisabilité** Le système devrait vérifier, pour toute demande d'accès, qu'elle est réalisable techniquement
19	**Journalisation et gestion des alarmes** Toutes les actions entreprises par les différents acteurs y compris SAGA devraient être enregistrées dans un journal sous forme d'événements. Certains de ces événements peuvent générer des alertes.
20	**Tests de fonctionnement** Le système devrait permettre de tester le bon fonctionnement d'un accès.

6.5.2 Exigences fonctionnelles non supportées par notre système

Le système SAGA tel que défini dans ce travail ne peut pas satisfaire les exigences suivantes :

1	**Traitement des demandes exceptionnelles** Certaines demandes sont exceptionnelles en raison de leur nature ou leur caractère unique. Dans ce cas, il est souvent nécessaire de réunir un certain nombre de personnes qui seront chargées de vérifier dans quelle mesure la demande met ou non en péril la sécurité globale de l'organisation.
2	**Traitement des demandes à effet immédiat** En cas d'urgence, un utilisateur peut demander un accès immédiat à certaines ressources déterminées, à condition qu'il le justifie. Dans ce cas, il n'y a pas de contrôle humain et l'intervention est réalisée directement par le système sur la ressource en question. C'est ce qu'on appelle une demande à effet immédiat.
3	**Configuration automatique des composants.** Il s'agit réaliser directement la configuration sur la ressource, sans intervention de son gestionnaire.
4	**Inventaire des accès réels aux ressources** Il s'agit d'établir la liste des ressources auquel une personne a accès.

6.6 *Contraintes non fonctionnelles*

1	**Rapidité** Le système devrait gérer toute demande d'accès en en minimisant le délai de traitement.
2	**Économie** Le système proposé devrait permettre de réaliser une économie des coûts de gestion des accès et une économie en temps.
3	**Simplification d'utilisation** Le système devrait apporter une certaine simplification dans la tâche des acteurs, notamment par : o l'automatisation de certaines tâches o la simplification et l'uniformisation des formulaires o une aide en ligne o des messages informatifs ou d'avertissement o la possibilité de cacher la complexité aux acteurs non avertis
4	**Sécurité** Toutes les actions entreprises par le système devraient se conformer aux règles de sécurité de l'organisation.
5	**Disponibilité et accessibilité** Le système devrait être disponible à tout moment et accessible à partir de tout endroit d'où les acteurs exercent leurs activités.
6	**Convivialité** L'interface du système devrait être intuitive, rapide et conviviale.
7	**Intégration dans le système informatique de l'organisation** Le système proposé devrait s'intégrer dans l'infrastructure informatique existante de l'organisation.
8	**Extensibilité** Le système devrait être extensible et pouvoir gérer par exemple de nouvelles ressources ou de nouveaux types de demande.
9	**Outil d'aide à la gestion des accès** Le système devrait procurer une aide efficace pour la gestion des accès sans toutefois suppléer à la décision humaine.
10	**Organisation fonctionnelle** Le système devrait tenir compte de l'aspect fonctionnel et de la structure de l'organisation
11	**Gestion centralisée des demandes d'accès de toute nature** Le système devrait prendre en charge, de manière centralisée, l'ensemble des demandes d'accès aux services qui sont disponibles dans l'organisation. Le système devrait alors être considéré comme un passage obligé.

12	**Exécution des tâches clé** Le système devrait permettre à chaque acteur de réaliser l'ensemble de ses tâches principales du point de vue de la gestion des accès.
13	**Généricité du système** Le système devrait pouvoir s'adapter aux multiples domaines d'application présents au sein d'une organisation.
14	**Communication facilitée** Le système devrait faciliter et rendre efficace la communication entre les différents acteurs.
15	**Cohérence des informations** Le système devrait garantir la cohérence des informations

6.7 *Traçabilité*

La traçabilité de l'étape de synthèse des exigences est exprimée dans le tableau ci-dessous.
En ligne, on y retrouve les

- exigences issues de l'étude de l'existant
- exigences issues de l'analyse des besoins
- exigences déduites de nos observations
- objectifs du système

En colonne, on retrouve le produit de la synthèse ; c'est à dire la répartition de ces exigences et objectifs en

- exigences fonctionnelles couvertes par SAGA
- exigences fonctionnelles non couvertes par SAGA
- contraintes non fonctionnelles

Les cases coloriées en rouge symbolisent le relation « devient ».

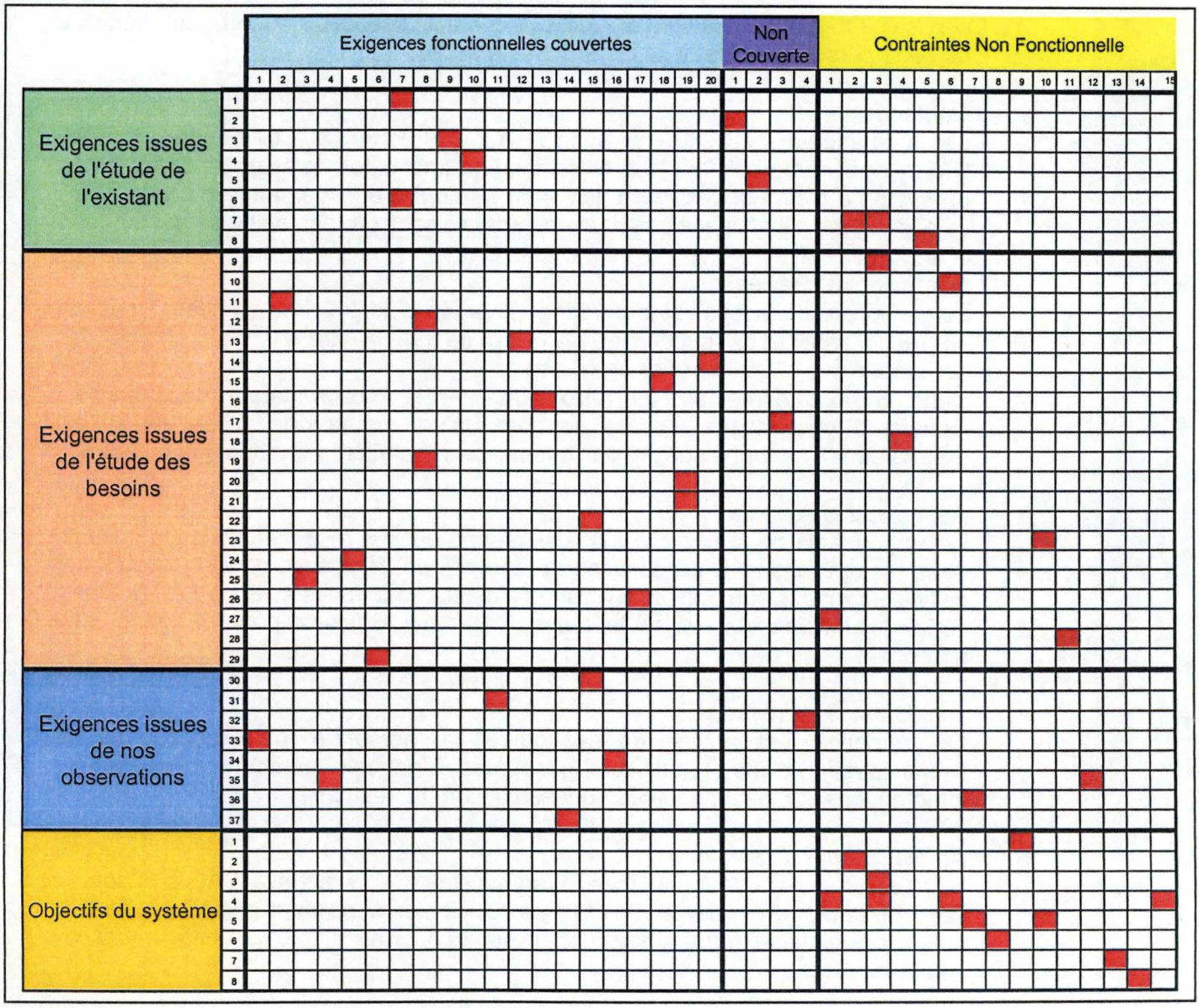

Figure 4 - Tableau de traçabilité de l'étape de synthèse des exigences

7 Définitions des concepts

Les définitions suivantes vont donner de manière précise la signification et la portée des termes employés pour décrire les concepts du système.

- *Personne*
  Dans notre système SAGA, il y a plusieurs acteurs humains que l'on regroupe sous le vocable « personne » et qui jouent différents rôles par rapport à celui-ci. Ils peuvent d'ailleurs parfois cumuler ces rôles. Nous avons recensé les différents rôles suivants :

  - *Utilisateur du système*
    Comme son nom l'indique l'utilisateur du système est une personne qui va utiliser le système SAGA pour poser différentes actions.

  - *Bénéficiaire*
    Le bénéficiaire est une personne susceptible de tirer profit d'un ou plusieurs services. On parle également d' « utilisation » d'un service. C'est au bénéfice de cette personne qu'est faite la demande d'accès.

  - *Demandeur*
    Le demandeur est la personne qui introduit la demande d'accès. Il faut cependant noter que des acteurs non-humains tels le système lui-même ou un système extérieur peuvent également introduire une demande d'accès. Dans ce cas, même s'il ne s'agit pas d'une personne, nous parlons aussi de demandeur.

  - *Gestionnaire d'accès*
    Le gestionnaire d'accès est la personne chargée d'administrer les accès. Il est de sa responsabilité de superviser, à l'aide de SAGA, la prise en charge des demandes ainsi que leur mise en œuvre.
    Dans certains domaines, on retrouve le terme « coordinateur décentralisé » lorsque le gestionnaire d'accès ne gère qu'une partie des accès qui lui est généralement attribuée suivant des critères de répartition géographique ou autre.

  - *Responsable sécurité*
    Le responsable de la sécurité est la personne qui est responsable de la sécurité dans l'organisation. Toute question de sécurité est réglée en dernier recours par cette personne. Cette personne n'est pas toujours acteur vis à vis de SAGA. Dans certains domaines on utilise le terme « Security Officer » pour désigner cette personne.

  - *Gestionnaire de ressource*
    On appelle gestionnaire de ressource la personne qui se charge de l'administration d'une ou plusieurs ressources. C'est cette personne qui réalise les interventions sur les ressources dont elle assure la gestion.

  - *Administrateur du système*
    La personne qui est chargée d'administrer le système SAGA s'appelle l'administrateur du système. Ses tâches principales consistent gérer les différents objets du système (profils, services, groupes, ...) ainsi que la configuration de celui-ci.

- *Groupe*
  Dans notre système, on retrouve plusieurs notions de groupes. Elles se différencient par la nature des membres qui composent ces groupes.

  - *Groupe d'utilisateurs du système*
    Ce groupe d'utilisateur est un ensemble d'utilisateurs du système. Cet agglomérat est utile dans le contexte de l'attribution d'un niveau de sécurité à un ensemble de personnes. Un niveau de sécurité détermine pour chaque utilisateur du système les actions qu'il sera autorisé à poser via le système.

  - *Groupe de bénéficiaires*
    Un groupe de bénéficiaires est un ensemble de personnes qui, au travers d'une demande d'accès, peuvent bénéficier d'un même accès. Cette notion est surtout utile pour éviter de devoir demander un même accès séparément pour chaque individu d'un groupe. Un groupe de bénéficiaires peut être, soit un groupe de personnes constitué pour la circonstance, soit correspondre à un groupe fonctionnel.

    - *Groupe fonctionnel*
      Un groupe fonctionnel est un ensemble de personnes qui jouent un même rôle fonctionnel au sein de l'organisation. Il s'agit par exemple d'une équipe de personnes, d'une division ou d'un bureau. Un utilisateur peut n'appartenir à aucun groupe fonctionnel. Il peut aussi appartenir à plusieurs groupes fonctionnels

  Il est à noter que dans SAGA, les groupes sont uniquement constitués de personnes et non de groupes ; ce qui pourrait dans certains cas, limiter ses capacités d'intégration avec la structure de l'organisation.

- *Ressource*
  Une ressource est un objet fonctionnel qui joue un rôle dans l'accès aux services et qui peut être configuré à cet effet. Sa principale caractéristique est son atomicité.

  La configuration de chaque ressource est assurée par un gestionnaire de ressources.
  Une ressource peut être de nature diverse (exemple : un router Cisco™, un emplacement de parking, firewall, un local, ...). Dans la plupart des domaines, il s'agira d'un objet physique ; ce n'est pas le cas dans le domaine des réseaux où il peut aussi bien s'agir d'une machine que d'un composant logiciel.

  La notion de ressource est étroitement liée à celle de chemin d'accès développé ci-dessous.

- *Chemin d'accès (CA)*
  Intuitivement, on peut présenter le chemin d'accès comme étant la suite des ressources par lesquelles il est nécessaire de transiter pour rejoindre, à partir d'un point d'accès, la ressource cible qui preste le service demandé. Nous pouvons illustrer cela par un exemple emprunté au domaine des parkings : un véhicule qui veut rejoindre la place de parking no 118 (la ressource cible), située au premier sous-sol, doit avoir l'autorisation de franchir la porte d'entrée du parking (le point d'accès) et doit présenter une carte d'accès à la barrière de contrôle du premier sous-sol (une ressource intermédiaire).

  Pour être précis, nous dirons qu'un chemin d'accès est la suite ordonnée des ressources sur lesquelles il faut intervenir (nous parlons de ressources *impactées*) pour donner l'accès à une ressource-cible donnée à partir d'un point d'accès donné.

  La première ressource d'un chemin d'accès s'appelle le « point d'accès ».
  La dernière ressource d'un chemin d'accès s'appelle la « ressource-cible ».
  Les éventuelles autres ressources du chemin d'accès s'appellent les « ressources intermédiaires ».

  Ces types de ressources correspondent en fait à des rôles joués par les ressources dans le chemin d'accès. Une ressource peut cumuler plusieurs rôles au sein de chemins d'accès différents ou au sein d'un même chemin d'accès (par exemple : être point d'accès et ressource-cible s'il n'y a qu'une seule ressource dans le chemin d'accès).

  - *Point d'accès*
    Le point d'accès est le point à partir duquel le bénéficiaire d'un accès va utiliser celui-ci. Cette notion se matérialise sous des formes les plus diverses en fonction du domaine dans lequel on se trouve. Dans le domaine des réseaux informatiques, on peut imaginer qu'il s'agit du poste de travail de l'utilisateur. Dans le domaine des parkings, il peut s'agir de la porte d'entrée ouest du parking. Il s'agit d'une ressource à part entière dans la mesure où le point d'accès peut devoir être configuré afin de permettre l'accès au service.
    Un chemin d'accès doit toujours avoir un et un seul point d'accès.

  - *Ressource-cible*
    Une ressource-cible est une fonctionnalité ou un objet dont il peut être fait usage et auquel on peut avoir accès. La ressource-cible est la dernière ressource d'un chemin d'accès. Dans notre système, on suppose que toutes les ressources-cible peuvent être accessibles. La ressource cible peut être également définie comme étant le composant matériel ou logiciel qui héberge l'objet qui rend le service dont l'utilisateur souhaite pouvoir faire usage. Un chemin d'accès doit toujours avoir une et une seule ressource-cible.

    Exemples de ressources-cible :
    - Application informatique (logiciel, serveur de fichier, boite mail, ..)
    - Locaux, emplacement de parking

  - *Ressource intermédiaire*
    Une ressource intermédiaire est une ressource qui se situe sur le chemin d'accès entre le point d'accès et la ressource-cible et qui joue un rôle dans la connectivité entre ces deux-ci. Il est à noter que certains chemins d'accès ne comportent pas de ressources intermédiaires. L'ordre de passage au travers de ces éventuelles ressources intermédiaires peut avoir de l'importance.

Le chemin d'accès prévoit aussi bien le cheminement aller que le cheminement retour entre un point d'accès et une ressource-cible. Dans de nombreux cas, ces cheminements seront identiques.

L'association d'une ressource à un chemin d'accès se fait au travers de la fonction que celle-ci remplit au sein de ce chemin d'accès. En effet, une ressource peut, selon les circonstances, jouer une ou plusieurs fonctions au sein d'un même chemin d'accès ou au sein de chemins d'accès différents. Dans le domaine des parkings, ce concept peut être illustré par l'exemple d'un ressource qui est une barrière (ressource intermédiaire). Cette barrière peut remplir, au sein d'un même chemin d'accès, la fonction de contrôle d'entrée (dans ce cas, elle est actionnée par un bouton poussoir) et celle de contrôle de sortie (dans ce cas, elle est actionnée par un ticket dûment payé).
Un autre exemple, dans le domaine des réseaux, est celui d'un logiciel de firewall (ressource intermédiaire) qui remplit dans un chemin d'accès la fonction de proxy et dans un autre chemin d'accès la fonction de router filtrant. Cette notion de fonction n'influençant en rien les mécanismes du système SAGA (dans son état actuel), nous avons décidé de ne pas la modéliser.

Dans notre modèle, il peut y avoir plusieurs chemins d'accès possibles pour une paire « point d'accès – ressource-cible » donnée. Voyez à ce sujet le chapitre 8.3.3 'Les chemins d'accès (CA)' qui traite de la détermination des chemins d'accès inhérents à un service pour une demande d'accès donnée.

- *Service*

Un service est un ensemble de ressources-cibles (au moins une) qui représente un intérêt pour l'utilisateur et auquel ce dernier peut demander accès. La notion de service a été introduite pour pouvoir présenter au demandeur des notions d' « objets auxquels on peut demander accès » qui ont une signification pour lui. Ceci est particulièrement utile dans les domaines hautement spécialisés (exemple : les réseaux informatiques).
Il est de la responsabilité de l'administrateur du système de définir harmonieusement les différents services auxquels il peut être demandé accès.

Dans certains cas (exemple : le parking) le service représente une seule ressource-cible (exemple : l'emplacement de parking) ; dans d'autres (exemple : accès de base au système informatique central) le service peut représenter une multitude de ressources-cibles.

Exemples de services :
- accès à Internet
- utilisation de la messagerie
- emplacement de parking dans un garage
- accès au coffre d'une banque, ...

- Lien entre la notion de Service et celle de Chemin d'Accès
  Pour connaître les ressources qui sont impactées lorsque l'on doit mettre en œuvre une demande d'accès à un service, il faut pouvoir déterminer les différents chemins d'accès qui sont concernés par ce service.
  C'est lors de la définition d'un service que l'on précise les différents chemins d'accès (liste exhaustive) qui sont liés à l'accès à un service. Il est important de remarquer qu'un chemin d'accès n'a pas lieu d'exister s'il n'est pas rattaché à un service.

  Le schéma ci-dessous illustre le rattachement des ressources à un service au travers des chemins d'accès. On notera les cas particuliers suivants :
  - la ressource 3 joue le rôle de point d'accès dans le CA 6 et le rôle de ressource intermédiaire dans le CA 8 ;
  - il y a plusieurs CAs possibles pour le Service A pour aller de la ressource2 à la ressource 6 ;
  - le CA 8 et le CA 1 passent par les mêmes ressources dans le même ordre mais concernent des services différents, il s'agit donc de CAs différents.

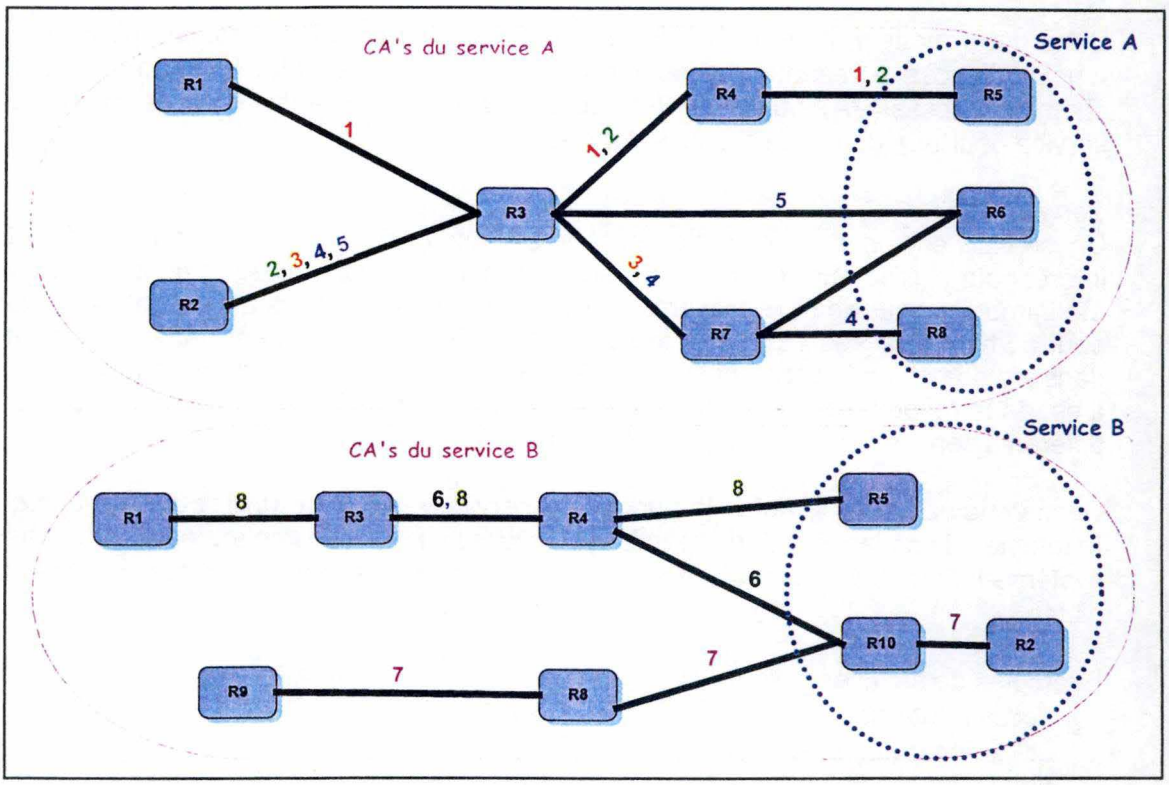

Figure 4 - Chemins d'accès / Service

Légende :

- Les ressources sont représentées par des R1
- Un chemin d'accès(CA) est représenté par une suite de ressources reliées par une ligne noire qui comporte un même numéro : R1 — 4 — R3
  Certains tronçons sont communs à plusieurs CAs, ils comportent donc plusieurs numéros. Pour rendre les CAs plus visibles, les numéros des tronçons d'un même CA ont été coloriés d'une même couleur.
- Les CAs relatifs à un service sont entourés d'un trait rose
- Un CA se lit de gauche à droite ; la ressource la plus à gauche d'un CA étant donc le point d'accès du CA et celle la plus à droite étant la ressource-cible du CA.
- Les ressources-cible relatives à un service sont entourées d'un trait pointillé bleu

- *Accès*
  Un accès représente le droit d'utiliser un service. Il ne faut pas confondre avec l'utilisation de l'accès qui constitue le fait d'utiliser ce droit d'accéder au service. Un accès est toujours donné à un bénéficiaire ou à un groupe de bénéficiaires.

- *Demande d'accès*
  Une demande d'accès consiste à demander d'appliquer une opération sur un accès pour un bénéficiaire ou un groupe de bénéficiaires donné.

  Cette *opération sur l'accès à un service* consiste soit en

  - la création d' un nouvel accès : consiste à ajouter pour ce bénéficiaire un accès au service demandé;

  - la modification d'un accès existant : consiste à modifier un accès que possède déjà le bénéficiaire au service demandé (exemple : prolongation d'un accès);

  - la suppression d'un accès existant : consiste à supprimer un accès pour ce bénéficiaire;

  - l'activation d'un accès existant : un accès peut exister sans toutefois (encore) être actif. On entend par actif le fait que le bénéficiaire peut l'utiliser. Cette nature de demande consiste à rendre actif un accès existant. Remarquons que la vérification qu'un accès est actif ou non lors de son utilisation ne fait pas partie des fonctionnalités de SAGA.

  - la désactivation d'un accès existant : lorsqu'un accès existe et qu'il est actif, on peut demander de le désactiver. Ceci est particulièrement intéressant dans le cas d'une interruption temporaire d'activité d'une personne ; cela permet de ne pas devoir recréer l'accès lors de la reprise d'activité.

Dans la suite du document, nous parlerons de « type de demande d'accès » plutôt que d' « opération sur l'accès à un service ». Nous dirons donc, par exemple qu'une demande d'accès est du type « création d'un nouvel accès ».

Pour rappel, celui qui introduit la demande d'accès s'appelle le demandeur.

Le bénéficiaire de cette demande peut être le demandeur lui-même ou une tierce personne pour laquelle le demandeur est habilité(=autorisé) à faire ce type de demande. De la même manière, le demandeur peut ne pas faire partie du groupe de bénéficiaires.

Différents contrôles sont appliqués sur une demande d'accès pour vérifier si celle-ci peut être acceptée par SAGA en regard des règles de sécurité en vigueur dans l'organisation – c'est le contrôle de légitimité – et pour vérifier si la demande est réalisable – c'est le contrôle de faisabilité.
Dans la plupart des domaines, il existe une notion de « demande à effet immédiat ».
C'est un type de demande pour lequel les contrôles sont réduits au minimum et sont automatisés (sans intervention humaine) de manière à ce que cette demande puisse être mise en œuvre immédiatement. Les demandes à effet immédiat sont généralement utilisées dans un contexte d'urgence mais dans des conditions d'application très strictes.

- *Intervention*
  L'intervention est l'ensemble des adaptations qu'il faut faire à la configuration d'une ressource pour mettre en œuvre la demande d'accès. Dans certains domaines, on parle de « définitions » à réaliser. Cette intervention est réalisée par le gestionnaire de la ressource.

  Voici quelques exemples d'interventions :
  - Création d'un compte
  - Routage
  - Filtrage
  - Modification d'une Access Control List
  - Translation d'adresses
  - Enregistrement d'un code d'accès
  - Attribution d'une carte d'accès
  - Paramétrisation d'une barrière
  - ...

  Cette notion ne fait pas partie des concepts utilisés par SAGA mais est intéressante dans le contexte d'une extension de SAGA dans laquelle la mise en œuvre automatisée des interventions sur les ressources serait possible.

- *Demande d'intervention*
  Une demande d'intervention définit l'action de demander à un gestionnaire de ressource de réaliser une intervention sur une ressource dont il assure la gestion. Dans notre système, une demande d'intervention se matérialise par un formulaire qui est envoyé au gestionnaire de ressource par tout moyen utile et qui contient toute l'information nécessaire à celui-ci pour effectuer son intervention. Dans SAGA, c'est le gestionnaire de ressource qui réceptionne les demandes d'intervention et qui est chargé de les mettre en œuvre.

- *Profil*
  Un profil correspond à un ensemble de services auquel une personne à accès sans devoir le demander et ce, dès la création de cette personne dans le système ou le rattachement de celle-ci à un groupe. A chaque personne ou à chaque groupe correspond un profil au plus.

  La liste des accès octroyés à une personne par ses profils est obtenue en combinant par un OU logique les accès qui lui sont octroyés par son profil personnel ainsi que ceux octroyé par le profil de tous les groupes auxquels elle appartient.

  Seul l'administrateur du système gère la composition des profils; il détermine donc quels sont les services qui sont liés à chaque profil. Il n'y a donc pas de notion de « demande » dans le système SAGA pour la gestion de la composition d'un profil.

  Exemple : un profil « nouvel arrivant » pourrait regrouper les services « Internet », « accès de base au mainframe » et « messagerie » et être associé à chaque personne nouvellement engagée.

- *Action*
  Dans le contexte de SAGA, nous appelons Action tout acte posé par un utilisateur du système au travers de SAGA.

- *Niveau de sécurité*
  Le niveau de sécurité est une caractéristique de utilisateur du système qui définit quel est son champ d'action au travers du système SAGA :
  - à chaque utilisateur du système correspond un niveau de sécurité ;
  - à chaque action possible correspond un niveau de sécurité requis.

  Il pourrait, par exemple, être décidé que seuls les personnes possédant le niveau de sécurité 10 peuvent modifier les profils des groupes.

- *Système*
  Dans ce travail, lorsque l'on parle de système, il faut entendre SAGA. Dans les autres cas, il est précisé de quel système il s'agit. Nous parlons également de système externe pour désigner un système d'information qui est externe à SAGA.

- *Autorité d'approbation*
  L'autorité d'approbation est une notion qui intervient de manière optionnelle dans le contrôle de légitimité(voir le chapitre 8.3.6 'Contrôle de légitimité' pour plus de détails). Il s'agit d'une liste de personnes ou de groupes qui doivent donner leur accords sur la légitimité d'une demande d'accès. Lorsqu'il s'agit d'un groupe, l'accord d'une seule personne du groupe suffit. On ne peut donc pas dire que l'autorité d'approbation est un groupe au sens du terme tel que défini ci avant.

- *Information de configuration*
  Lors d'une demande d'accès, il est primordial de recueillir du demandeur des informations qui permettront plus tard au gestionnaire de ressource de mettre en œuvre les demandes d'intervention découlant de cette demande d'accès. Nous employons le terme « information de configuration » pour désigner ces informations. Pour permettre cela, il est nécessaire d'associer à chaque ressource les informations de configuration (par exemple : l'adresse IP de la station de travail, le numéro de carte d'identité du demandeur, la taille du véhicule, ...)qu'il convient de recueillir et ce pour chaque type de demande d'accès possible. En effet, nous considérons que les informations de configuration à recueillir pour une demande de création d'un accès ne sont pas nécessairement les mêmes que celles qu'il faut recueillir pour la suppression de cet accès, même s'il s'agit d'une même ressource.

- *Mutation*
  On dit qu'il y a mutation d'une personne lorsque celle-ci change de groupe fonctionnel. Seules des personnes peuvent muter.

8 Analyse du système

8.1 Introduction

L'objectif général de ce chapitre est, en partant des exigences émises par les différents acteurs et de la liste des concepts fondamentaux du système d'information de SAGA , d'en modéliser les données et les traitements.
Les différents produits de cette analyse sont :

- le modèle des données présenté selon le formalisme *ERA* (Entity Relationship Association)
- la liste des scénarios de traitement et des différentes actions de ceux-ci représentés selon le formalisme des diagrammes de séquence *UML*
- la liste des modules fonctionnels de SAGA et des différentes fonctions de ces derniers

8.2 Modèle des traitements

8.2.1 Introduction

L'utilisation de scénarios reflète une image assez réelle du fonctionnement d'un système. De plus, leur compréhension est relativement aisée pour un public non averti. Ils permettent ainsi de décrire dans sa globalité le fonctionnement du système. C'est pourquoi nous allons, dans ce chapitre, dresser la liste des scénarios de traitement de SAGA. Nous avons choisi de détailler parmi ceux-ci le scénario décrivant la demande de création d'un accès qui, selon nous, est le plus représentatif en terme de taille et de complexité.

8.2.2 Méthode utilisée

Dans un premier temps, la liste des exigences émises par les différents acteurs nous permettra de dresser la liste des scénarios de traitement de SAGA. Ensuite nous détaillerons le scénario générique décrivant la demande de création d'un accès. Nous découperons ce scénario en étapes qui contiendront différentes actions. Nous définirons, sur base des propos recueillis lors de l'enquête, l'enchaînement de ces actions en le représentant sous forme de diagrammes de séquence UML.

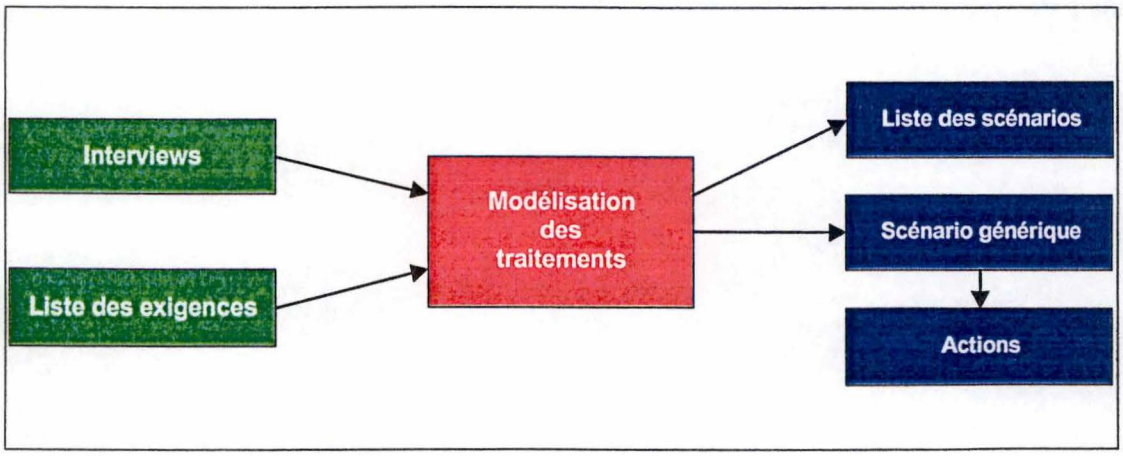

Figure 5 - Modélisation des traitements

Légende :
Les rectangles verts contiennent les informations nécessaires pour la modélisation
Le rectangle rouge indique la phase d'analyse
Les rectangles bleus indiquent les informations produites par l'analyse

8.2.3 Liste des scénarios

La liste ci-dessous reprend tous les scénarios significatifs pour la description du système SAGA.

- Authentification d'un utilisateur dans le système
- Demande d'accès (création/modification/suppression/activation/désactivation)
- Gestion des chemins d'accès (création/modification/suppression/ajout ou suppression d'une ressource)
- Consultation des informations du système et rapports
- Contrôle des accès aux fonctionnalités du système
- Gestion des personnes (création/suppression/modification/mutation)
- Gestion des groupes (création/suppression/modification/ajout d'une personne/suppression d'une personne)
- Gestion des profils (création/suppression/modification/ajout d'un service/suppression d'un service)
- Gestion des services (création/suppression/ajout d'un chemin d'accès/suppression d'un chemin d'accès)
- Gestion des alarmes
- Gestion de la journalisation des évènements (audit)
- Gestion des accès échus ou inutiles
- Importation d'informations (reprise de l'existant)
- Exportation d'informations
- Gestion des menus (affichage/sélection)

8.2.4 Scénario générique d'une demande d'accès

Une demande d'accès peut se décliner en cinq types d'opération différents :

- *Création d'un nouvel accès*
  L'objectif de cette opération consiste à configurer l'accès à un service pour un bénéficiaire.

- *Suppression d'un accès existant*
  Consiste à défaire ce qui a été entrepris dans une opération de création définie ci-dessus, c'est à dire supprimer la configuration entreprise pour cet accès.

- *Modification d'un accès existant*
  Permet la modification d'un paramètre lié à un accès à un service existant, par exemple, le prolongement d'un accès temporaire.

- *Activation d'un accès existant*
  Cette opération permet d'ouvrir l'accès à un service qui aura été configuré auparavant grâce à une opération de création.

- *Désactivation d'un accès existant*
  La désactivation permet d'empêcher l'utilisation d'un service par un bénéficiaire sans devoir modifier la configuration de cet accès.

Dans certains domaines d'application, les opérations d'activation et de désactivation sont intégrées respectivement dans l'opération de création et de suppression.

Une demande d'accès peut provenir de trois sources différentes :

- *Une personne physique*
  Il s'agit de la voie classique, une personne effectue une demande d'accès via l'interface interactive du système.

- *Une application externe*
  Le système permet à une application externe d'effectuer une demande d'accès. On peut citer comme exemple, le cas d'une application de gestion des ressources humaines qui effectue une demande d'accès pour un nouveau membre du personnel.

- *Le système lui-même*
  Il est parfois nécessaire que le système effectue des demandes d'accès pour maintenir la cohérence de la liste des accès existants. On peut citer comme exemple, la modification d'un chemin d'accès ayant pour conséquence la reconfiguration de toutes les ressources concernées. Dans ce cas, c'est donc le système qui initie les demandes d'accès.

Le scénario générique du traitement d'une demande d'accès décrit ci-dessous, a été découpé en quatre étapes pour une question de clarté :

- Étape 1 : Saisie de la demande
- Étape 2 : Contrôles
- Étape 3 : Interventions
- Étape 4 : Clôture

Chaque étape comprend une série d'actions entreprises soit par le système, soit par des acteurs extérieurs au système. Ces actions peuvent être spécifiques suivant le domaine d'application. Pour simplifier, on considère dans ce scénario que chaque action se déroule normalement et se termine avec succès.

Dans la suite de ce chapitre, nous allons décrire en détail le scénario d'une demande de création d'un accès à un service introduite par une personne physique. Nous y préciserons les actions entreprises par le système lors du déroulement de ce scénario.

8.2.4.1 Étape 1 : Saisie de la demande

Contexte

Nous considérons que dans cette étape, les scénarios d'authentification (identification du demandeur par le système) et de sélection (demande d'accès/création d'un nouvel accès pour un service) dans le menu se sont correctement déroulés.
Le demandeur a également fourni au système les informations suivantes :

- la référence du bénéficiaire;
- la référence du service souhaité.

Sur base de ces informations, le système détermine les ressources impactées et génère un formulaire demandant toutes les informations nécessaires pour la configuration de ces ressources. Ce formulaire est alors présenté au demandeur.

Diagramme de séquence UML

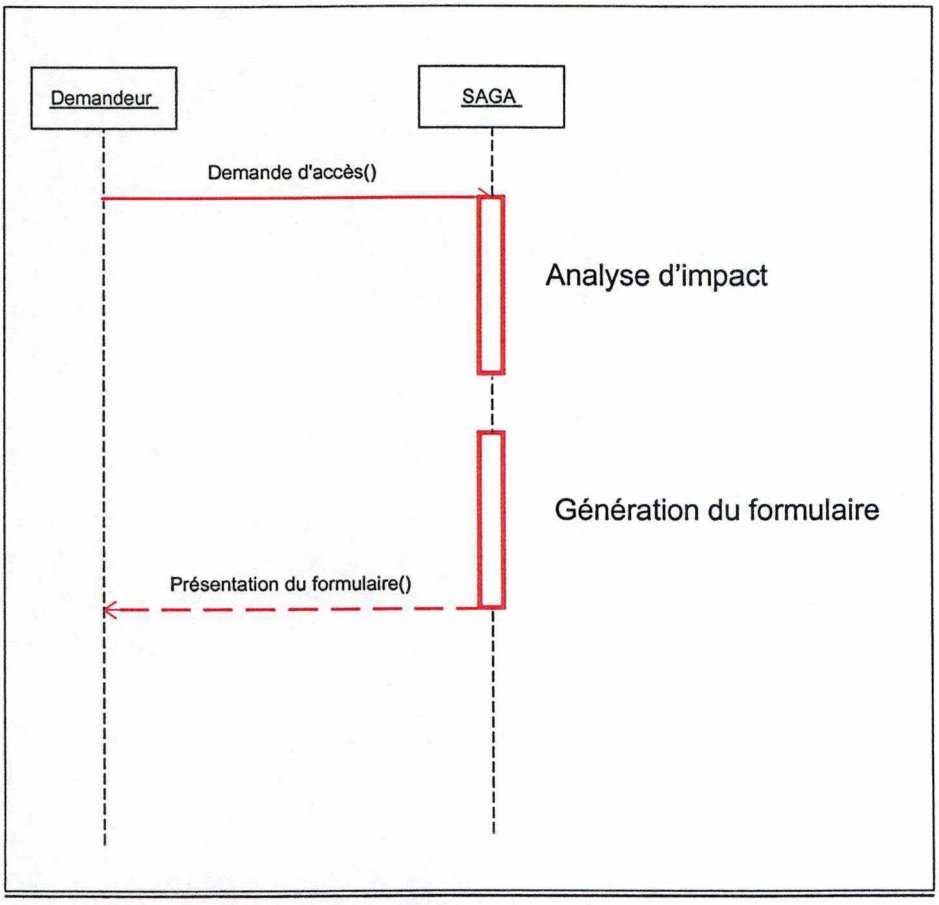

Figure 6 - Diagramme de séquence UML – étape 1

Actions entreprises par le système

- *Analyse d'impact*
  Cette action consiste à déterminer les chemins d'accès, et par conséquence les ressources, qui sont impactés par la demande d'accès.

- *Génération du formulaire*
  Les informations à fournir par le demandeur peuvent varier suivant les ressources impactées par cette demande. Chaque ressource nécessite un certain nombre d'informations pour sa configuration. Cette action doit effectuer une consolidation des informations requises et éliminer les éventuelles informations redondantes. L'ensemble de ces informations constitue le formulaire qui sera présenté au demandeur.

8.2.4.2 Étape 2 : Contrôles

Contexte

Le demandeur complète le formulaire et le renvoie au système. Le système effectue une série de vérifications sur les informations contenues dans le formulaire telle qu'un contrôle de validité des données, un contrôle de légitimité et un contrôle de faisabilité technique.

Diagramme de séquence UML

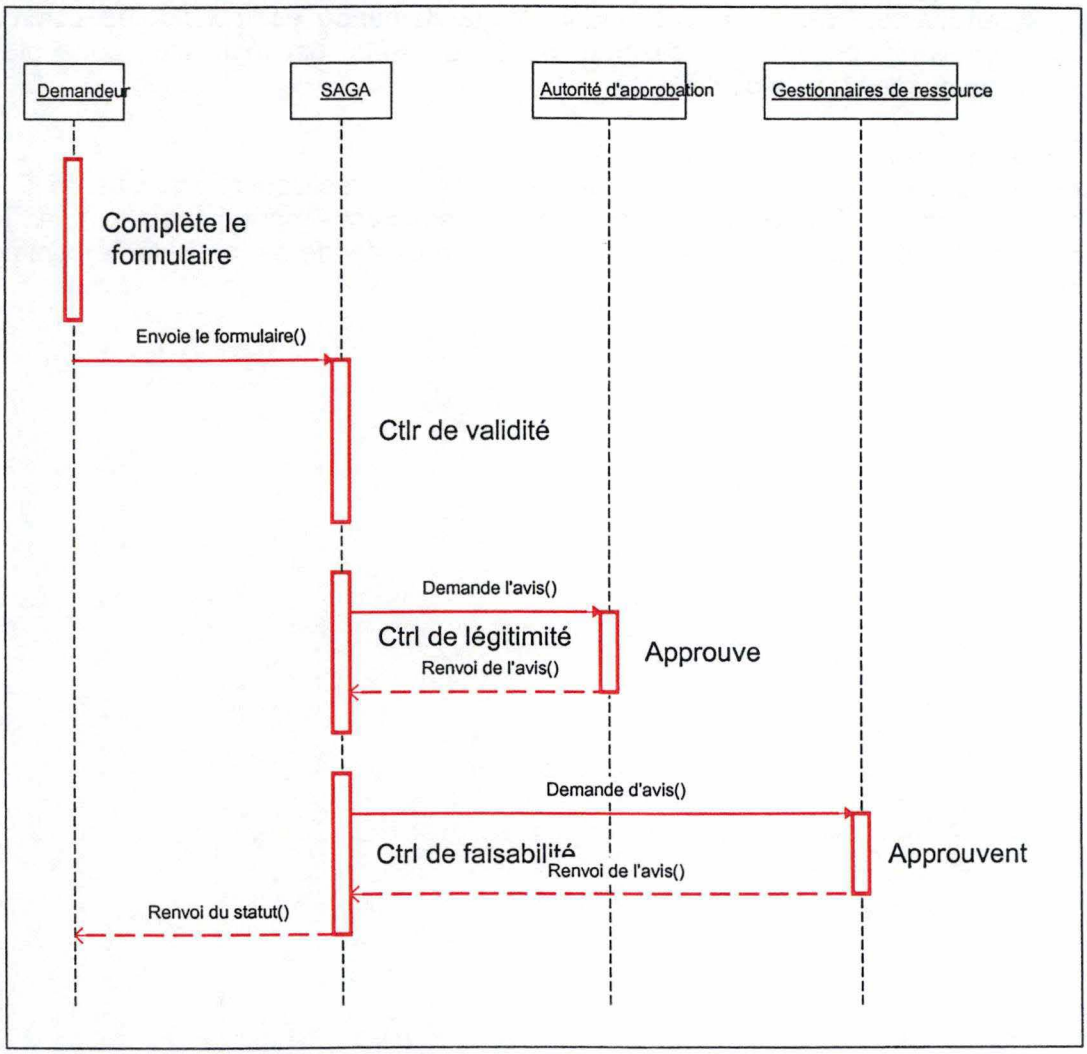

Figure 7 - Diagramme de séquence UML – étape 2

Actions entreprises par le système

- *Contrôle de validité des informations saisies*
  Ce contrôle consiste à vérifier la pertinence des informations saisies par le demandeur.

- *Contrôle de légitimité*
  Ce contrôle vérifie si le demandeur est habilité à effectuer ce type de demande et si le bénéficiaire peut utiliser le service demandé. Ces contrôles sont étroitement liés aux règles d'octroi d'un accès en vigueur au sein de l'entreprise. Ce contrôle peut être automatique, c'est à dire effectué par un programme ou manuel et dans ce cas, il appartient à un groupe de personnes (l'autorité d'approbation) de donner son accord. Dans ce dernier cas, s'il n'y a pas unanimité, la demande est considérée comme illégitime. Veuillez consulter le chapitre 8.3.6 'Contrôle de légitimité' pour plus d'informations sur ce type de contrôle.

- *Contrôle de faisabilité*
  Ce contrôle a pour but de vérifier si une demande est réalisable techniquement. Il consiste à contacter les gestionnaires dont les ressources sont impactées par le traitement de la demande. S'il n'y a pas unanimité, la demande est considérée comme non faisable.

8.2.4.3 Étape 3 : Intervention

Contexte

L'ensemble des contrôles effectués dans l'étape précédente est considéré comme positif. Dans cette étape, des demandes d'interventions seront générées par le système et transmises aux gestionnaires de ressources impliqués.

Diagramme de séquence UML

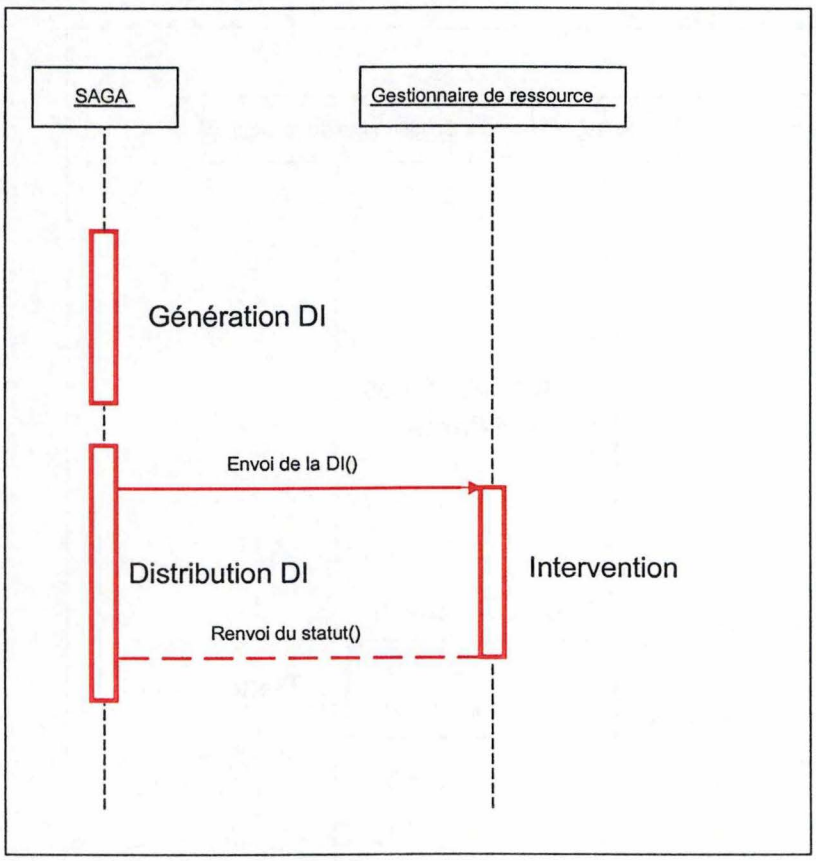

Figure 8 - Diagramme de séquence UML – étape 3

Actions entreprises par le système

- *Génération des demandes d'interventions (DI)*
  Il s'agit de la phase de traduction de la demande d'accès en autant de demandes d'interventions qu'il existe de ressources impactées par cette demande. Ces formulaires de demande d'interventions contiennent les opérations et les informations nécessaires à la configuration de chaque ressource.

- *Distribution DI*
  Chaque demande d'intervention est transmise au gestionnaire de ressource correspondant. Ce dernier renvoie, après l'intervention, une réponse indiquant le statut de celle-ci (positif ou négatif).

8.2.4.4 Étape 4 : Clôture

Description

Les gestionnaires de ressources concernés par cette demande ont confirmé que les demandes d'interventions ont été effectuées avec succès. Dans cette étape, la liste des accès existants sera mise en concordance avec les modifications effectuées. Ensuite, la disponibilité de l'accès demandé est signalée au demandeur ainsi qu'au bénéficiaire et des tests sont proposés pour vérifier le bon fonctionnement de l'accès nouvellement créé.

Diagramme de séquence UML

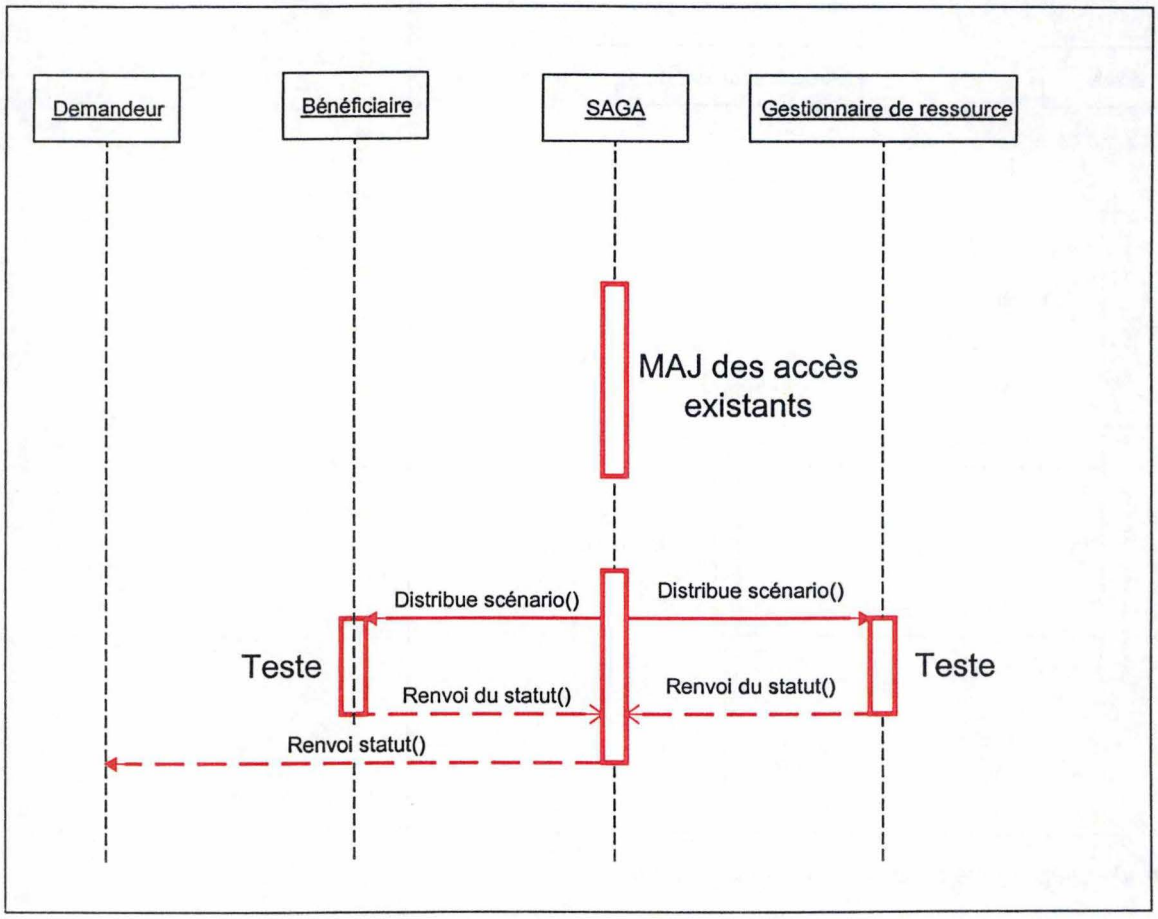

Figure 9 - Diagramme de séquence UML – étape 4

Actions entreprises par le système

- *Mise à jour de la liste des accès existants*
  Dés que l'étape précédente s'est déroulée avec succès, le système effectue la mise à jour de la liste des accès existants afin que celle-ci reflète la nouvelle situation.

- *Vérification du bon fonctionnement de l'accès*
  Il est dans certains cas opportun de tester l'accès accordé. Dans ce cas, le système offre la possibilité d'exécuter un scénario de test préétabli. Ce scénario peut impliquer le demandeur et les gestionnaires de ressources. Ce scénario devra être défini pour chaque service.

- *Retour d'informations*
  Cette action consiste à avertir le demandeur et le bénéficiaire du statut de cette demande. Dans le cas de la demande pour la création d'un nouvel accès, un mode d'emploi peut éventuellement être joint à cette confirmation. Cette confirmation peut être transmise soit de manière électronique (ex : courrier électronique) soit manuelle (courrier).

8.3 Discussion des choix fondamentaux

8.3.1 Introduction

Vous trouverez dans ce chapitre les justifications et les raisonnements qui ont mené aux choix fondamentaux issus des questions apparues lors de l'analyse et la conception de SAGA. Ces choix sont discutés autour d'une série de questions auxquelles nous tentons de donner réponse soit en comparant différentes alternatives, soit en proposant des pistes de réflexion à approfondir. Pour nous guider dans notre démarche, nous avons essayé de garder à tout moment à l'esprit les exigences exprimées lors de l'analyse des besoins et plus particulièrement les contraintes non fonctionnelles qui en découlent. Il y a bien sûr d'autres questions auxquelles nous avons du trouver réponse mais, compte tenu du caractère limitatif d'un mémoire, nous avons choisi d'en présenter ici uniquement les principales. Les réponses à ces autres questions se retrouvent dans les choix exprimés tout au long de l'analyse et de la conception du système SAGA.

8.3.2 Communication entre SAGA et les gestionnaires de ressources

8.3.2.1 Contexte

Les gestionnaires de ressources sont des acteurs importants du système SAGA. En effet c'est par leur intervention sur les ressources qu'est matérialisée la mise en œuvre des demandes d'accès. Il est donc important de trouver un moyen efficace de communiquer avec ces acteurs. Les différents aspects de cette problématique sont discutés ci-après autour de trois questions. Les deux premières questions concernent l'échange d'information de SAGA vers les gestionnaires de ressources : la première discute du contenu tandis que la seconde discute du moyen de communication. La troisième question aborde le problème du retour d'information des gestionnaires vers SAGA.

8.3.2.2 Comment générer le formulaire de demande d'accès (FDA) présenté au demandeur ?

Le formulaire de demande d'accès doit récolter les informations permettant aux gestionnaires de ressources de configurer correctement celles-ci pour mettre en œuvre une DA à un service déterminé. Pour rencontrer la CNF no 3 (*simplicité)*, le FDA doit être compréhensible et simple à utiliser par le demandeur. Pour rencontrer les CNFs no 6 (*convivialité*) et 14 (*faciliter la communication*), nous avons décidé de présenter un seul FDA par service demandé et non pas un FDA par ressource impactée.
Lors de la création d'un service, le gestionnaire de chaque ressource impactée est invité à donner la liste des informations de configuration qui lui sont nécessaires pour configurer un accès à ce service sur sa(ou ses) ressource(s) et ce pour chaque type de demande d'accès possible.
Pour établir le formulaire de demande, il faut donc rassembler chacune de ces informations et les présenter au demandeur. Il se peut cependant qu'il y ait des redondances parmi ces informations : par exemple, pour configurer la ressource 5 et la ressource 18 il se peut qu'faille le numéro de carte d'identité du bénéficiaire.
Ces redondances proviennent essentiellement du fait qu'une même information de configuration peut être requise pour plusieurs ressources différentes. Les contraintes nous interdisent cependant de le demander plus d'une fois au demandeur la même information ; notre système doit donc effectuer une consolidation des informations requises pour chaque ressource impactée et éliminer les redondances éventuelles. La difficulté est de trouver une technique qui permette de présenter un formulaire duquel on a éliminé les informations redondantes.

Dans ce contexte, nous avons envisagé les alternatives suivantes :

Alternative 1 :
Lors de la création d'un service, tous les gestionnaires de ressources sont réunis et créent un FDA pour ce service, en tenant compte du type de demande d'accès et des informations de configuration de chaque ressource impactée.

Avantages :
- Aucun traitement ne doit être fait pour générer le FDA lors de l'introduction de la DA (gain de temps)
- Relativement facile à mettre en place lors de la création du service car les gestionnaires sont déjà réunis pour fournir la liste des informations de configuration pour chaque ressource du CA

Inconvénients
- Nécessite de réunir les nombreux gestionnaires de ressources pour avoir un FDA commun
- Doit être adapté lors de chaque modification de la composition d'un CA lié au service
- Les chemins d'accès sont figés lors de la création du service (pas de choix au moment du traitement de la DA)

Alternative 2 :
Plutôt que de réunir les avis de tous les gestionnaires de ressources, SAGA présente lors de la création d'un service une ébauche de FDA au gestionnaire d'accès qui devra ensuite le compléter et supprimer les redondances. Cette ébauche de FDA est générée dynamiquement sur base des informations de configuration des ressources composant le service.

Avantages :
- aucun traitement ne doit être fait pour générer le FDA lors de l'introduction de la DA
- facile à mettre en place lors de la création du service
- seul le gestionnaire d'accès intervient, pas les gestionnaires de ressources

Inconvénients
- requiert l'intervention du gestionnaire d'accès
- nécessite la gestion des modifications de la composition des CAs : à chaque fois une nouvelle ébauche du FDA doit être représentée au gestionnaire d'accès
- les chemins d'accès sont figés lors de la création du service (pas de choix au moment du traitement de la DA)

Alternative 3 :
SAGA offre un référentiel des types d'informations requis par les ressources. Le gestionnaire de ressource choisi dans ce référentiel les informations qu'il désire recueillir du demandeur. Le gestionnaire d'accès gère ce référentiel et en assure l'intégrité. Grâce à ce système, SAGA peut éliminer facilement les redondances lors de la création du FDA en repérant simplement les informations qui ont la même identification dans le référentiel.

Avantages :
- génération dynamique du FDA : s'adapte aux changements de composition des CAs
- pas d'intervention humaine (économie de temps et de ressources humaines)
- convient également aux domaines dans lesquels la composition des CAs est fort changeante
- peut servir à un contrôle de validité des informations vis à vis des domaines de valeur exprimés dans le référentiel
- réutilisation du référentiel pour les informations de configuration à fournir par rapport au service en général

Inconvénients

- il faut gérer le référentiel de manière centralisée et s'assurer que chaque information de celui-ci n'est pas « sémantiquement » le double d'une autre (fastidieux) ; pour éviter cela, une définition claire de chaque terme doit être fournie (assez complexe)
- il faut éduquer les gestionnaires de ressources à l'utilisation du référentiel ; ce qui n'est pas toujours facile car ils ont chacun leur propre jargon dans leur domaine

Choix :
Notre choix s'est porté sur l'alternative 3 parce que, bien que plus difficile à mettre en œuvre, elle offre une bonne adaptabilité en cas de modification de la composition d'un CA . C'est également l'alternative qui rencontre le mieux la CNF no 3 *(cacher la complexité)* car elle permet, lorsqu'il y a une bonne gestion du référentiel, d'utiliser toujours le même terme pour désigner une même chose ; ce qui facilite fortement la compréhension et l'apprentissage.
Remarquons que pour des domaines où les CA sont peu nombreux et où leur composition est très stable, l'alternative 1 serait probablement le meilleur choix, mais pas en toute généralité. C'est le cas dans le domaine des parkings où l'on emprunte toujours le même chemin pour accéder aux emplacements et où le nombre de chemins est quasi figé dès le début.

8.3.2.3 Distribution des DI's

Dans SAGA, la distribution des DI's aux gestionnaires de ressources est assurée par un mécanisme de *workflow* (WF). Ceci a comme principal avantage que l'on peut utiliser des scénarios qui consistent à faire circuler des documents à une liste prédéfinie de destinataires.

La liste des destinataires du WF de distribution des DI's est constituée du gestionnaire de ressource de chaque ressource impactée par la DA.

Ce choix arrêté, il nous restait à voir comment nous allions pouvoir concrétiser au mieux la CNF no 7 qui demande que le système s'intègre au mieux avec l'existant et avec les contraintes de fonctionnement de l'organisation. La question est donc voir comment on peut interfacer SAGA avec un système de WF existant ou dans le cas contraire en mettre un en œuvre.

Comment communiquer avec un système de workflow existant ?

L'idéal serait de pouvoir « sous-traiter » les workflows à un produit existant. Cela exige que ce produit puisse importer les formulaires et les scénarios et être initié au départ de SAGA.
L'intégration de notre système dans un environnement existant implique qu'il devrait pouvoir interagir avec des logiciels de workflow de différentes natures (Lotus Notes,...) déjà présents dans la société. Cependant, il n'existe actuellement pas de standard définissant l'interfaçage d'applications avec des logiciels de workflow.
On peut tout de même retrouver sur le marché quelques tentatives de normalisation(OMDA, MAPI-WF,WAPI). Voyez à ce sujet le site [http://63.33.34.189/library/middleware.shtml].

Notons que la « WorkFlow Management Coalition » (WFMC)) soutient WAPI (workflow application interface) et propose plusieurs recommandations permettant à une application d'exporter dynamiquement vers un logiciel de workflow des scénarios comprenant les rôles et les acteurs. Voyez à ce sujet le site [http://www.aiim.org/wfmc/mainframe.htm].

Conclusions :
Malgré que le WorkFlow ne fait pas partie du noyau de SAGA, il soutient tout de même trois parties importantes du système :

- la communication avec les différents acteurs impliqués dans le processus de contrôle d'une demande d'accès (légitimité et faisabilité)
- la distribution des demandes d'intervention
- les tests de fonctionnement de l'accès

Il est donc important qu'il s'intègre parfaitement avec les systèmes de communications interpersonnels qui existent déjà dans l'organisation. Nous considérons cependant que concevoir un système de WF pour SAGA dépasse largement le cadre de ce travail. Nous conseillons donc d'utiliser le logiciel de WF qui existe déjà dans l'organisation lorsque c'est possible. Cela évitera, de devoir former les utilisateurs à un nouvel outil. L'interfaçage de SAGA avec ce logiciel de WF devra se faire au cas par cas.

8.3.2.4 Retour d'information

Une des exigences exprimées lors de l'analyse des besoins est de fournir à chaque fois que c'est possible de l'information au demandeur et au bénéficiaire sur l'état d'avancement du traitement de la demande. Le traitement des demandes d'intervention par les gestionnaires de ressources est un moment important dans le traitement d'une DA ; il convient donc de donner de l'information en retour à cette occasion.

A ce sujet, nous avons arrêté les principes suivants :

- le retour d'information est envoyé uniquement au demandeur et au(x) bénéficiaire(s) ; les autres acteurs peuvent obtenir cette information par consultation de SAGA.
- la forme du retour d'information est commune à tous les acteurs (tous reçoivent donc la même information).
- le retour d'information est généré après réception d'un rapport d'exécution de la part des gestionnaires de ressources, sur chaque DI distribuée.
- il y a un seul retour d'information par DA

8.3.3 Les chemins d'accès (CA)

8.3.3.1 Comment déterminer les CA's à prendre en compte pour mettre en œuvre une DA ?

Sans vouloir revenir sur la définition d'un CA que vous pouvez retrouver au chapitre 7 'Définitions des concepts', on peut présenter le CA comme étant la suite des ressources par lesquelles il est nécessaire de transiter pour rejoindre, à partir d'un point d'accès, la ressource cible qui preste le service demandé.
Dans les multiples domaines que veut couvrir SAGA, il existe une multitude de manières de déterminer quel chemin emprunter pour rejoindre telle ressource à partir de tel point.
La seule connaissance du point d'accès et de la ressource cible ne suffit pas.

Prenons deux exemples à titre d'illustration :

- dans le domaine des parkings on peut imaginer qu'un véhicule (le bénéficiaire) qui veut accéder la place numéro 112 (ressource cible) à partir de l'entrée nord (point d'accès) doive emprunter un chemin (CA) différent en fonction de différents paramètres : son poids, sa taille, le type d'abonnement, l'heure qu'il est, ...

- dans le domaine des réseaux, on peut facilement imaginer qu'il y ait plusieurs routes (CA) pour accéder un serveur de fichier (ressource cible) à partir d'un poste de travail (point d'accès). Il peut y avoir une route principale et des routes alternatives qui ne sont activées qu'en cas de panne ou lors de travaux d'entretien. Remarquons que dans SAGA, nous ne nous intéressons qu'aux ressources sur lesquelles une intervention est nécessaire pour mettre en œuvre l'accès au service.

Il faut distinguer le choix du chemin emprunté au moment de l'utilisation de l'accès et le choix du chemin d'accès au moment de définir l'accès. Il va de soi que l'accès accordé doit pouvoir être utilisé, ce qui veut dire que la définition de l'accès doit couvrir toutes les possibilités d'utilisation de cet accès. Le principe de SAGA est donc de tout prévoir, de manière à ce que tous les chemins possibles soient configurés. Plusieurs CAs concernant la même ressource-cible peuvent donc devoir être configurés.

On peut donc en déduire que le choix d'un chemin d'accès dépend d'une combinaison plus ou moins complexe de plusieurs critères, à savoir :
1. des attributs qualifiant le bénéficiaire
2. des attributs qualifiant le demandeur
3. des attributs qualifiant le service demandé
4. des informations de l'environnement d'exécution (connues au moment de l'utilisation de l'accès)

Au moment du traitement de la DA, tous ces éléments sont connus a priori sauf les informations de l'environnement d'exécution. Lors du traitement de la DA, il faut donc considérer tous les cas possibles du critère 4 dans les limites des critères 1 à 3. Il s'agit donc d'anticiper toutes les valeurs du critère 4. Pour ce faire, il est important que la règle qui permet de sélectionner le(s) CA(s) lors du traitement de la DA soit en phase avec celle qui permet de sélectionner le chemin à parcourir lors de l'utilisation de l'accès. Prenons par exemple le cas d'un camion qui, pour rejoindre sa destination, peut emprunter deux chemins différents sur lesquels il faut configurer des points de contrôle pour le laisser passer. L'idée exprimée ci-dessus suggère de définir ces deux chemins d'accès lors de la création du service et de les configurer tous les deux.

Tout ceci est extrêmement complexe à généraliser, c'est pourquoi nous avons choisi de confier la sélection des CAs lors du traitement de la DA à un module externe qui doit être adapté au cas par cas.

Ce choix nous permet d'offrir aux utilisateurs de SAGA une grande flexibilité dans l'élaboration des règles de sélection des CAs – en effet on peut considérer qu'un langage de programmation permet de mettre en œuvre les règles les plus simples aux plus compliquées – le prix à payer est essentiellement le manque de convivialité de cette option et sa complexité de mise en œuvre. Pour aplanir ce problème nous proposons deux autres alternatives, parmi lesquelles le gestionnaire d'accès peut choisir lors de la configuration du service dans SAGA, à savoir :

- une liste fixe des chemins d'accès
  Les chemins d'accès sont fixes, c'est à dire communs à toutes les demandes d'accès pour ce service. Ils sont déterminés lors de la mise en place du service.

- une liste variable des chemins d'accès lors de la mise en place du service
  Il s'agit donc de sélectionner m chemins d'accès parmi n possibles (m <= n). La sélection est accomplie grâce à une requête dynamique basée sur la valeur de ces paramètres.

8.3.3.2 Comment attacher les ressources impactées au service correspondant ?

Pour pouvoir donner accès à un service, il faut effectuer un certain travail de configuration sur des ressources, que nous appelons ressources *impactées*. Par ailleurs, nous avons défini que les ressources qui prestent le service s'appellent ressources-cibles et que le bénéficiaire de l'accès exerce celui-ci à partir d'une ressource appelée point d'accès. Nous avons donc décidé d'appeler « chemin d'accès » la liste des ressources impactées (et uniquement celles là) qui permettent de joindre la ressource-cible à partir du point d'accès. Nous associons ensuite à un service tous les chemins d'accès qui contiennent les ressources-cibles qui prestent ce service et, par voie de conséquence, toutes les ressources à configurer pour y donner accès.

- *Pourquoi ne considérer que les ressources « impactées » dans le chemin d'accès ?*

Au sein d'un CA, nous ne considérons que les ressources impactées parce qu'il n'y a que celles là qui nous intéressent dans le contexte de la problématique de gestion des accès. Si nous étions dans la peau d'un gestionnaire réseau, nous serions concernés par l'ensemble des ressources du réseau. Il en va de même pour le comptable du parking qui doit connaître tous les emplacements qui existent au sein du parking. Ce n'est pas le cas avec SAGA.

Ce point de vue nous prive de la possibilité d'avoir dans SAGA une représentation fidèle de l'ensemble des ressources de l'organisation. SAGA ne peut pas aider quelqu'un qui chercherait à faire un tel inventaire. Cela prive également SAGA de la possibilité de collaborer facilement avec un quelconque système dans lequel l'ensemble des ressources serait connu et géré, comme il en existe dans certains domaines.

Ce choix se justifie essentiellement dans le fait que la gestion des ressources ne fait pas partie des objectifs de SAGA. Lier cette problématique à celle de la gestion des accès cumulerait les difficultés.

Nous pensons cependant que la représentation cartographique des ressources d'une organisation et la gestion de ces ressources constituent deux extensions à SAGA qu'il serait intéressant d'étudier dans le cadre d'un prolongement à ce travail.

- *Doit-on vraiment définir, pour un service donné, un chemin d'accès pour chaque point d'accès ?*

Une des limites que présente actuellement SAGA en matière de convivialité est qu'il y est nécessaire de définir des chemins d'accès différents pour chaque couple « point d'accès – ressource cible ». Cela ne pose pas de problème dans les domaines où il y a peu de points d'accès (les parkings) ou peu de ressources cibles (les coffres d'une banque). Par contre dans les domaines complexes comme celui des réseaux informatiques, c'est une limite très gênante. Si, par exemple pour le service X, à partir de 1.000 stations de travail (point d'accès) des utilisateurs doivent pouvoir accéder à un serveur de fichier (ressource cible), il faudrait définir en SAGA 1.000 chemins d'accès différents.

Pour remédier à ce problème nous avons pensé qu'il serait intéressant de pouvoir regrouper les ressources au sein de groupes. Cela nous donnerait des groupes de ressources cibles, des groupes de point d'accès et pourquoi pas des groupes de ressources intermédiaires. Ces regroupements ne doivent se faire que par les gestionnaires du système dans leur propre intérêt (par exemple : pour limiter le nombre de CAs, pour faire coïncider la notion de groupe avec celle de « *subnet* », pour faire un regroupement géographique, ...).
Pour pouvoir rester dans les limites d'un mémoire, nous n'avons pas poussé plus en avant cette réflexion mais il serait intéressant de voir quelles sont les contraintes introduites par ces « groupes de ressources » dans nos modèles des données et traitements.

8.3.4 Comment générer les demandes d'interventions à partir de la demande d'accès ?

La finalité principale du système SAGA est de pouvoir mettre en œuvre des demandes d'accès. Pour cela, le système distribue à chaque gestionnaire de ressource concerné une demande d'intervention. Nous allons montrer ici comment nous avons conçu la déduction des demandes d'intervention à partir de la demande d'accès.

A chaque demande d'accès (DA) correspond autant de demandes d'intervention (DIs) qu'il n'existe de ressources impactées par cette demande. Il y a une DI par ressource. Chaque ressource impactée joue un rôle dans le fonctionnement global du service requis (connu au travers du CA). La configuration d'une ressource permettant à celle-ci de jouer ce rôle nécessite un ensemble d'informations qui sont fournies soit par le système soit par le demandeur. Le module de génération des demandes d'intervention a pour fonction de consolider – c'est à dire rassembler par ressource – ces informations et de les transmettre au gestionnaire de ressource.

La demande d'intervention contient, pour une ressource :
- toutes les informations de configuration fournies par le demandeur via le formulaire de demande (FDA) ou par le système
- les informations sur le service demandé
- le type de demande d'accès (création, modification, suppression, activation, désactivation)
- la référence de la ressource à configurer

Nous considérons qu'avec ces renseignements, le gestionnaire de ressources est capable de déduire l'intervention exacte qu'il doit faire ou ne pas faire sur la ressource.

8.3.5 Gestion des événements et de leurs impacts sur les accès

Si SAGA veut être l'outil unique de gestion des accès, il faut qu'il puisse donner une aide efficace aux gestionnaires d'accès. Nous pensons donc que SAGA doit permettre d'estimer les impacts sur les accès que peuvent avoir certains événements qui surviennent sur des objets connexes du système.
Nous tentons ici d'en faire une liste complète, tout en restant conscients qu'il existe des impacts indirects difficiles à détecter.
Une simple modification de la valeur d'un attribut d'une personne ou d'un autre objet de SAGA peut avoir comme conséquence des impacts sur les accès de cette personne.
Par exemple, si la localisation du bénéficiaire entre dans les critères qui déterminent le choix des CAs lors du traitement de la DA, alors une modification de la valeur de cet attribut peut remettre en cause le choix des CAs effectué à ce moment et donc nécessiter une nouvelle étape d'analyse d'impact et probablement de la génération de nouvelles DIs.
Pour optimiser le fonctionnement de SAGA, nous avons choisi de traduire chaque fois que c'est possible les impacts en DAs ; dans ce cas, SAGA peut jouer le rôle de demandeur en introduisant une DA. Ces DAs sont ensuite traitées par le même algorithme que celle introduites par des personnes.

D'autre part, il nous paraît clair que la collaboration de tous les gestionnaires (accès et ressources) est nécessaire pour pouvoir bien estimer les conséquences de chaque événement sur les accès existants. Peut-être serait-il intéressant d'étudier, en guise de prolongement à notre travail, la possibilité d'avoir un système expert d'analyse d'impact.

En attendant, nous avons arrêté les principes de base suivants :

- SAGA doit pouvoir produire un appel à un module externe pour chaque événement ci-dessous.
- Ce module externe est un module de programmation dans lequel les impacts sont codés. Le but est de générer les demandes d'accès (DAs) nécessaires afin de remettre la situation des accès en concordance avec la nouvelle situation de SAGA après l'événement.
- Si la gestion de certains objets est déléguée à un système externe, alors celui-ci devra pouvoir d'une manière ou d'une autre fournir ces événements à SAGA.

Liste des événements ayant des impacts sur les accès :

1. Objet « Personne »

   1.1 Action de « suppression »
   Il s'agit de supprimer une personne du système SAGA. Cette suppression entraîne généralement la suppression totale de tous ses accès, que cette personne les ait obtenus directement ou bien au travers de ses profils ou des groupes auxquels elle appartient.

   1.2 Action de « mutation »
   On parle de mutation d'une personne lorsque celle-ci change de groupe fonctionnel. Pour en déterminer les impacts, il faut d'abords répondre aux questions suivantes :
   - y a-t-il héritage optionnel des droits d'accès du groupe pour tout nouveau membre ? ou bien est-ce systématique ?
   - en cas de retrait d'un membre du groupe, quels accès doit-on supprimer pour cet ex-membre ?

   Il faut aussi faire attention aux impacts en cascade : une personne qui change de groupe change probablement aussi de profil de groupe.

1.3 Action d' « acquisition de son profil »
Une personne peut, lorsqu'elle n'en a pas encore, acquérir un profil. De ce fait, elle reçoit les accès liés à ce profil.

1.4 Action de « suppression de son profil »
Il s'agit de supprimer le profil d'une personne. De ce fait, cette personne perd probablement les accès qu'elle avait acquis grâce à ce profil.

1.5 Action de « changement de son profil »
Comme une personne n'a qu'un et un seul profil, elle peut en changer. Dans ce cas, nous assimilerons ce changement à une suppression de profil suivie d'une acquisition de profil.

2. Objet « Groupe »

2.1 Action de « suppression »
Il s'agit de supprimer un groupe existant dans SAGA. Cette suppression entraîne probablement la suppression des accès que les membres du groupe avaient acquis de part leur appartenance au groupe ou de par le profil de ce groupe.

2.2 Action d' « ajout d'une personne au groupe »
Il s'agit d'ajouter un membre à un groupe existant. En toute logique ce membre hérite des accès de ce groupe et du profil de ce groupe. Mais y-a-t-il héritage optionnel des droits d'accès du groupe pour tout nouveau membre ? ou bien est-ce systématique ?

2.3 Action de « suppression d'une personne du groupe »
Il s'agit de supprimer un membre du groupe. Ceci entraîne probablement la suppression des accès que cette personne avait acquis en rejoignant le groupe.

2.4 Action de « acquisition d'un profil »
Un groupe peut, lorsqu'il n'en a pas encore, acquérir un profil. De ce fait, tous les membres du groupe reçoivent les accès liés à ce profil.

2.5 Action de « suppression de son profil »
Il s'agit de supprimer le profil d'un groupe. De ce fait, tous les membres du groupe perdent probablement les accès qu'ils avaient acquis grâce à ce profil.

2.6 Action de « changement de son profil »
Comme un groupe n'a qu'un et un seul profil, il peut en changer. Dans ce cas, nous assimilerons ce changement à une suppression de profil suivie d'une acquisition de profil.

3. Objet « Profil »

3.1 Action de « suppression »
Il s'agit de supprimer de SAGA un profil existant. De ce fait, toutes les personnes qui avaient ce profil perdent les accès qui y sont liés. Tous les membres des groupes ayant ce profil perdent également les accès qui y sont liés.

3.2 Action d' « ajout d'un service »
Le profil donne accès à un service supplémentaire. De ce fait, toutes les personnes qui ont ce profil reçoivent un accès à ce service. Tous les membres des groupes ayant ce profil reçoivent également un accès à ce service.

3.3 Action de « suppression d'un service »
Le profil ne donne plus accès à ce service. De ce fait, toutes les personnes qui ont ce profil perdent leur accès à ce service. Tous les membres des groupes ayant ce profil perdent également leur accès à ce service.

4. Objet « Service »

4.1 Action de « suppression »
Il s'agit de supprimer un service dans SAGA ; ce qui veut dire qu'on ne peut plus y demander accès. Il y a des impacts évidents sur les accès déjà accordés à ce service via SAGA. Il faut probablement les annuler tous. SAGA va donc générer une série de demandes d'accès de type « suppression » pour tous les bénéficiaires d'accès à ce service et ensuite, lorsqu'il n'y aura plus personne qui y a accès, supprimera le service.

4.2 Action d' « ajout d'un chemin d'accès »
Il s'agit d'ajouter un nouveau chemin d'accès à un service. Les personnes qui ont déjà accès à ce service directement ou via un groupe auquel ils appartiennent, doivent probablement faire l'objet d'intervention sur les ressources de ce nouveau chemin d'accès. SAGA devra donc générer des DIs pour cette ressource pour chaque personne concernée.

4.3 Action de « suppression d'un chemin d'accès »
Il s'agit ici de supprimer un chemin d'accès qui est associé à un service. Ce n'est pas parce que l'on supprime un chemin d'accès que les ressources de celui-ci disparaissent du système. Il faut donc supprimer de ces ressources les configurations qui avaient été réalisées suite à des demandes d'accès impliquant ce chemin d'accès. SAGA va donc générer des demandes d'intervention pour annuler ces configurations. Le but étant toujours que la situation en SAGA reflète le mieux possible la situation au sein de l'organisation (et donc ici au sein des ressources).

5. Objet « Chemin d'accès »

5.1 Action de « suppression »
Normalement, on ne peut pas supprimer un CA de manière isolée ; dès qu'on le dissocie d'un service, il est supprimé. Il s'agit donc du même événement que le 4.3.

5.2 Action d' « ajout d'une ressource »
Il s'agit d'ajouter une ressource dans un chemin d'accès existant. Un chemin d'accès n'existe que s'il est rattaché à un service. Si on y ajoute une ressource, on doit probablement y faire une intervention pour les personnes ayant accès à ce service. SAGA va donc générer des demandes d'intervention.

5.3 Action de « suppression d'une ressource »
Il s'agit de supprimer une ressource dans un chemin d'accès existant. Un chemin d'accès n'existe que s'il est rattaché à un service. Si on en supprime une ressource, on doit probablement y faire une intervention pour les personnes ayant accès à ce service afin d'annuler la configuration pour ces personnes sur cette ressource. SAGA va donc générer des demandes d'intervention.

8.3.6 Contrôle de légitimité

Le contrôle de légitimité d'une demande d'accès comporte deux volets : le contrôle vérifiant si le demandeur est habilité à effectuer une telle demande et le contrôle consistant à vérifier si on peut accorder au bénéficiaire le service demandé. Ce contrôle peut être effectué de différentes façons. Lors de l'ajout d'un nouveau service, on précise quel sera le comportement à adopter pour le contrôle de la légitimité d'une demande d'accès. Il existe 3 comportements standards :

- Pas de contrôle de légitimité
  Aucun contrôle n'est effectué, toutes les demandes sont alors considérées comme légitimes

- Contrôle effectué par une autorité d'approbation
  L'autorité d'approbation est composée d'une liste fixée à l'avance de personnes ou de groupes qui doivent donner leur accord. Lorsque le membre d'une autorité d'approbation est un groupe, l'accord d'un seul membre du groupe est suffisant. L'accord des membres de l'autorité d'approbation est recueilli via un système de workflow. La composition de l'autorité d'approbation est définie pour chaque service. La liste des membres de l'autorité d'approbation peut être complétée avec le responsable du service, le coordinateur de sécurité du bénéficiaire et/ou son responsable hiérarchique.

- Contrôle effectué par un module externe
  Il est possible dans certains cas que le contrôle ne nécessite pas seulement l'approbation de personnes, mais qu'il soit basé sur un mécanisme de règles s'exécutant sur des informations spécifiques. Dans ce cas, un module externe, par exemple un script, permet d'effectuer le contrôle de légitimité. Nous supposons que toutes les informations nécessaires à ce contrôle sont fournies par la demande d'accès. Lorsque ce contrôle implique des personnes, le module externe doit pouvoir initier un workflow.

8.3.7 Problématique de la suppression des accès

Il y a dans SAGA cinq possibilités pour une personne d'obtenir un accès à un service :
1. en introduisant une demande d'accès pour ce service au bénéfice de cette personne
2. en se voyant attribuer un profil personnel qui donne cet accès
3. en introduisant une demande d'accès pour ce service au bénéfice d'un groupe dont fait partie cette personne
4. en faisant partie d'un groupe dont le profil donne cet accès
5. en changeant de groupe

Il se peut ainsi qu'il y ait des redondances dans la liste des accès existants d'une personne à un service car celle-ci pourrait avoir bénéficié de plusieurs possibilités parmi les cinq reprises ci-dessus; en effet une personne peut avoir un accès à un service à la fois par son profil personnel et parce que le groupe auquel elle appartient possède également cet accès. Dans ce cas, nous parlons d'accès redondants.

Il y donc également cinq possibilités de perdre un accès à un service :
1. en introduisant une demande de suppression d'accès pour cette personne
2. en changeant de profil personnel
3. en introduisant une demande de suppression d'accès pour un groupe dont fait partie cette personne
4. en faisant partie d'un groupe dont le profil change
5. en changeant de groupe

Les conflits surviennent lorsque SAGA reçoit, pour une personne, une demande ou un événement qui induit la suppression d'un accès à un service alors que celle-ci possède des accès redondants à ce même service.
Prenons un exemple : une personne peut avoir un accès à un service à la fois par son profil personnel et parce que le groupe auquel elle appartient possède également cet accès. Imaginons une demande de suppression d'accès pour cette personne. Doit-on supprimer l'accès ? Si oui, quel accès doit-on supprimer ?

Quelle attitude adopter vis à vis de ce problème ?

- Un point de vue est de dire que dans ce cas-ci, comme cet accès n'a jamais été accordé de manière individuelle (via une DA), il n'y a pas lieu de le supprimer via une DA individuelle. Cela signifie que l'on exige pour supprimer un accès que la demande corresponde à l'opération inverse de la demande qui avait permis l'octroi de cet accès. C'est possible mais dans ce cas la personne pourrait toujours avoir accès au service par un autre accès redondant.

- Un autre point de vue est de dire que, peu importe comment est demandée cette suppression, elle doit avoir lieu car cela est considéré comme une injonction. Il survient alors le problème d'inconsistance dans le système. En effet, si le groupe G à accès au service S, alors tous les membres du groupe G doivent avoir accès au service S, sauf Monsieur X pour qui il y a eu une DA spécifique de suppression de l'accès. Cela complique fortement les choses au moment où il faut, par exemple, estimer l'impact d'une mutation de Monsieur X du groupe G vers le groupe H.

- Un autre point de vue est de fixer des priorités (hiérarchiser) sur les 5 manières d'acquérir un accès et sur les 5 manières de s'en séparer. L'idée est que si on supprime un accès acquis par une manière qui a la priorité 4, on doit supprimer tous les autres accès à ce même service, acquis par une manière qui a une priorité inférieure ou égale. Cette suppression peut se faire de deux façons différentes :
  - suppression en cascade pour garder la cohérence. Si on doit supprimer un accès qu'une personne a via un groupe, on doit la sortir de ce groupe mais dans ce cas, elle perd les autres accès qu'elle avait via ce groupe.
  - suppression par injonction mais on retrouve le problème des inconsistances.

- Une autre idée serait de présenter au demandeur la situation d'accès telle qu'elle est et lui signaler que s'il veut supprimer complètement l'accès d'une personne à un service, il doit aussi faire le nécessaire au niveau des groupes et des profils en prenant contact avec le gestionnaire d'accès.

Nous avons choisi de laisser le choix au responsable de la sécurité qui pourra sélectionner lors de la configuration du système le principe qui colle le mieux avec celui en vigueur dans l'organisation.

8.4 Modèle des données

Nous présentons ici un modèle des données partiel qui se base sur la liste des concepts, la liste des acteurs et qui tient compte des aspects abordés au cours de la modélisation des traitements restreinte au scénario de traitement d'une demande de création d'accès.

Comme ce modèle des données est beaucoup trop touffu que pour être présenté de manière lisible en un seul tenant, nous avons choisi de le découper en plusieurs parties qui correspondent à des facettes différentes de la problématique générale.

Les différentes facettes du système sont :
- Accès
- Demande d'accès
- Chemin d'accès
- Ressources
- Légitimité
- Sécurité

Le formalisme utilisé pour représenté le modèle des données est celui du modèle Entité-Association tel que produit à partir du logiciel DB-Main(*)

8.4.1 Facette Accès

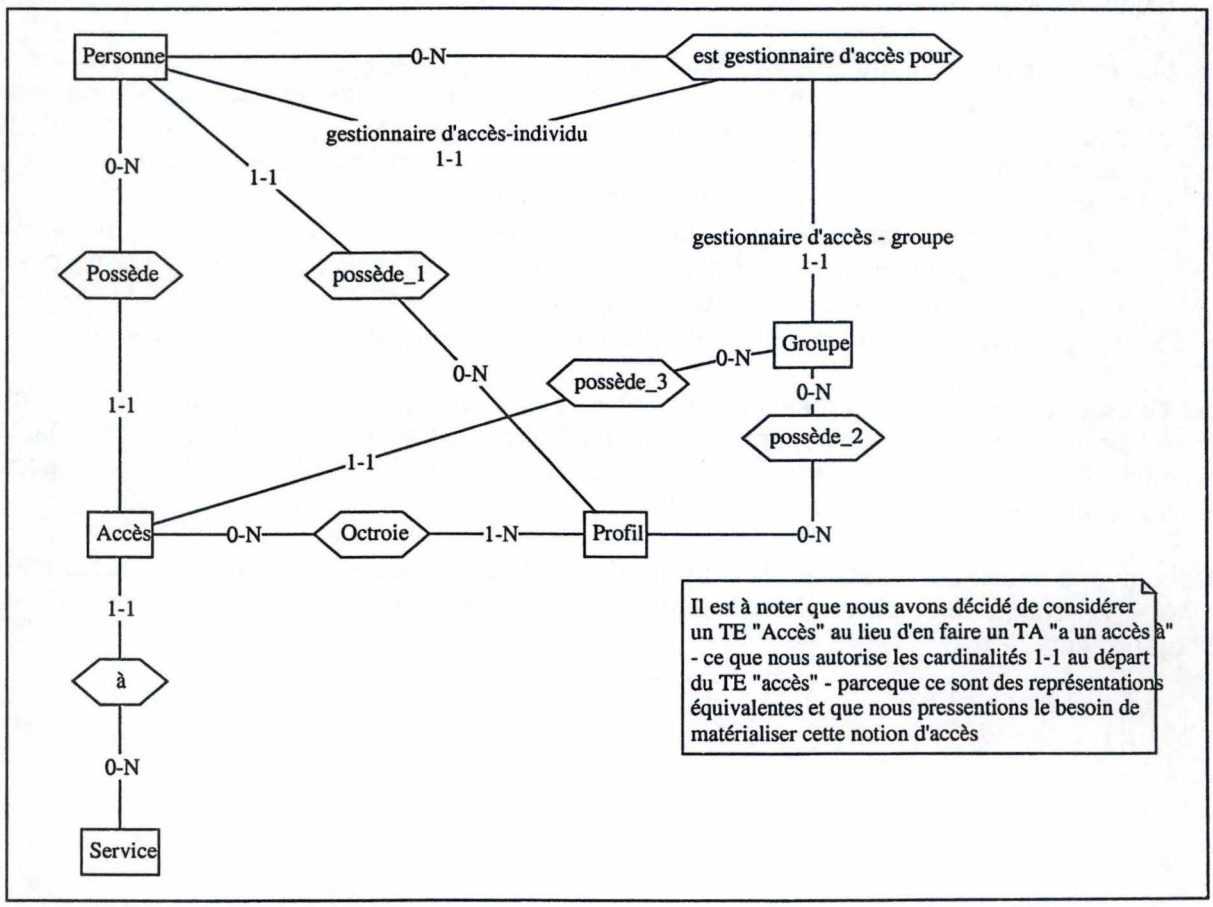

Figure 10 - Modèle des données - Facette "Accès"

(*) DB-MAIN is a research, development and technology transfer program developed by the Computer Science Institute of the University of Namur.

8.4.2 Facette Demande d'accès

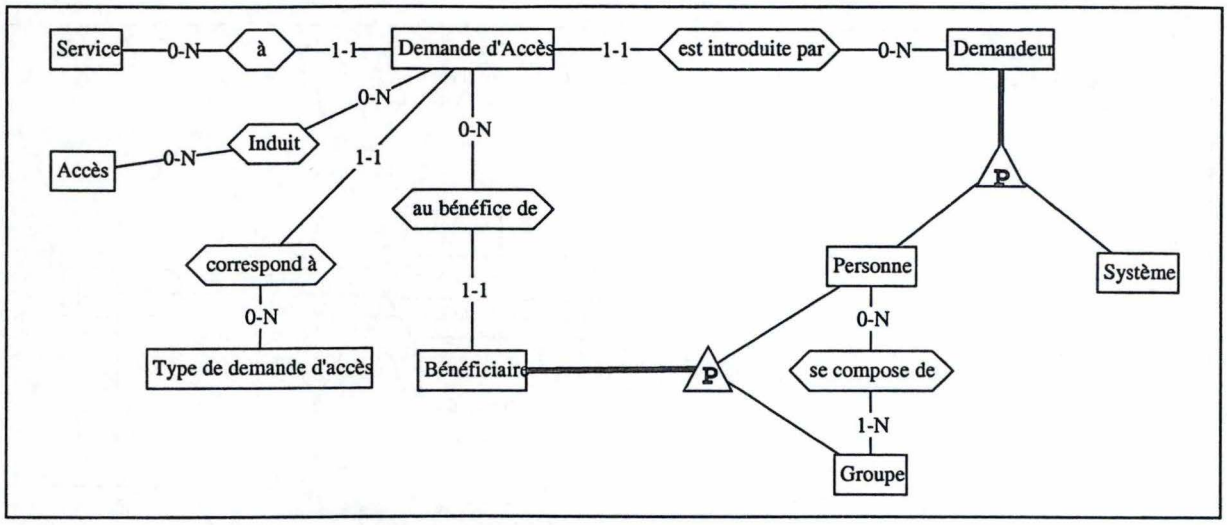

Figure 11 - Modèle des données - Facette "Demande d'accès"

8.4.3 Facette Chemin d'accès

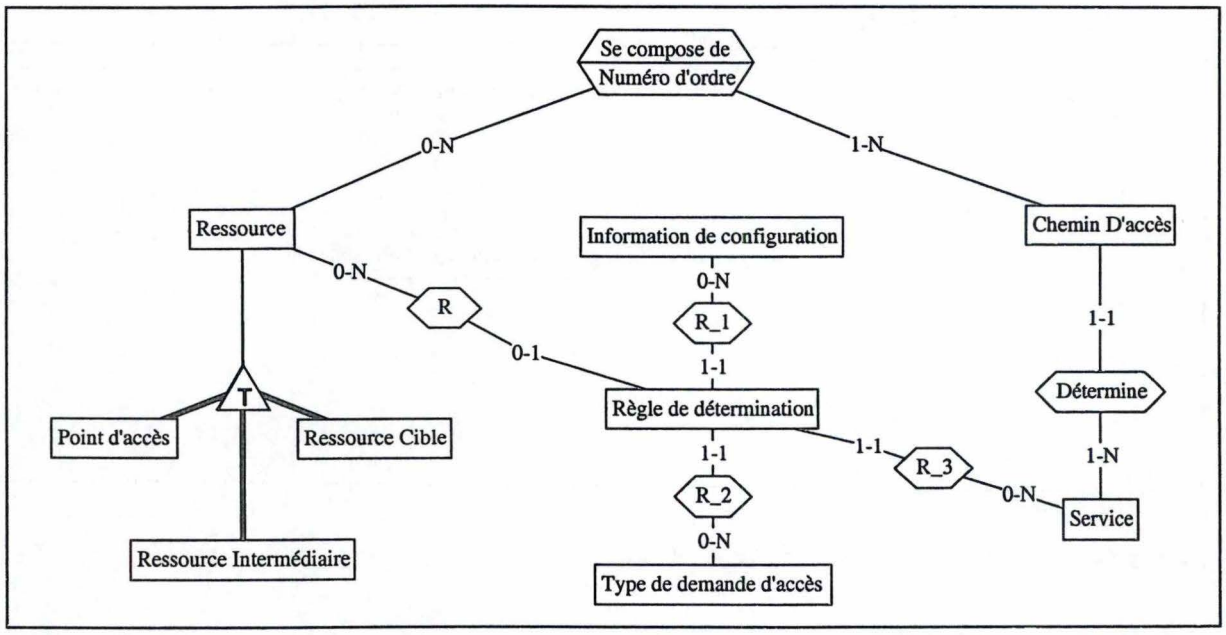

Figure 12- Modèle des données - Facette "Chemin d'accès"

8.4.4 Facette Ressource

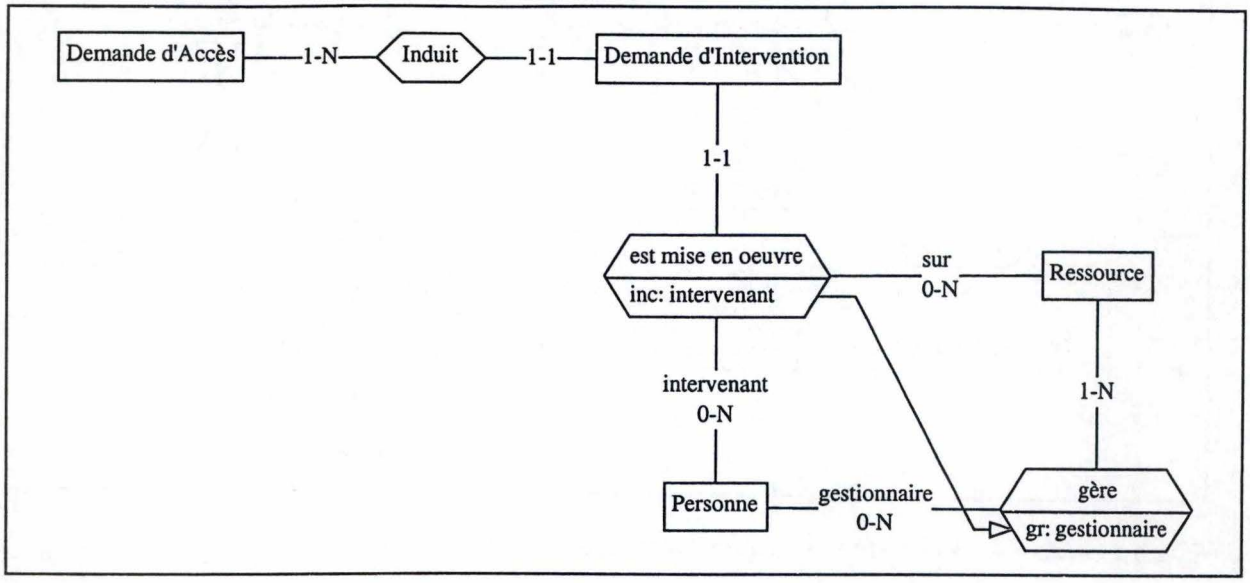

Figure 13 - Modèle des données - Facette "Ressource"

8.4.5 Facette Légitimité

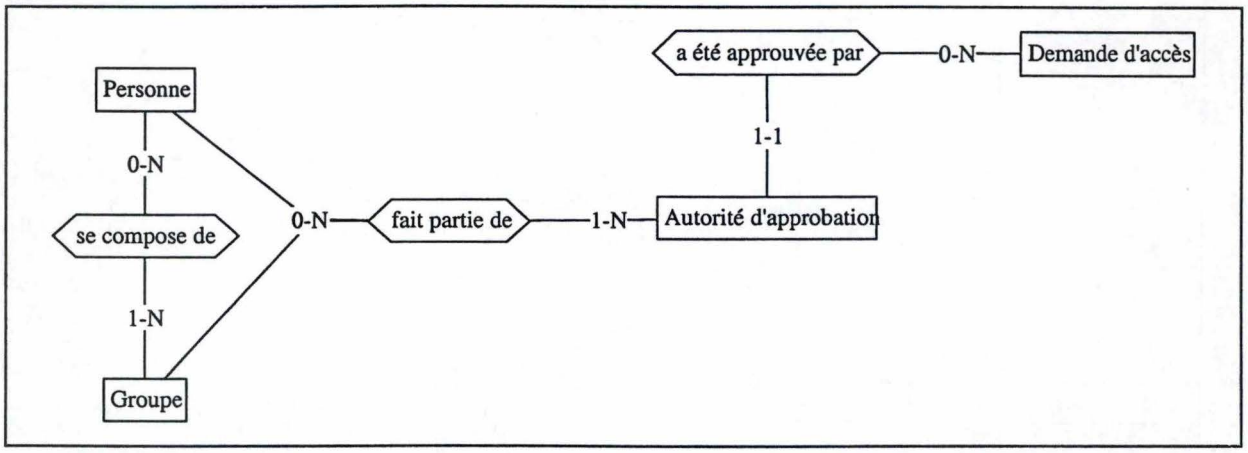

Figure 14 - Modèle des données - Facette " Légitimité"

8.4.6 Facette Sécurité

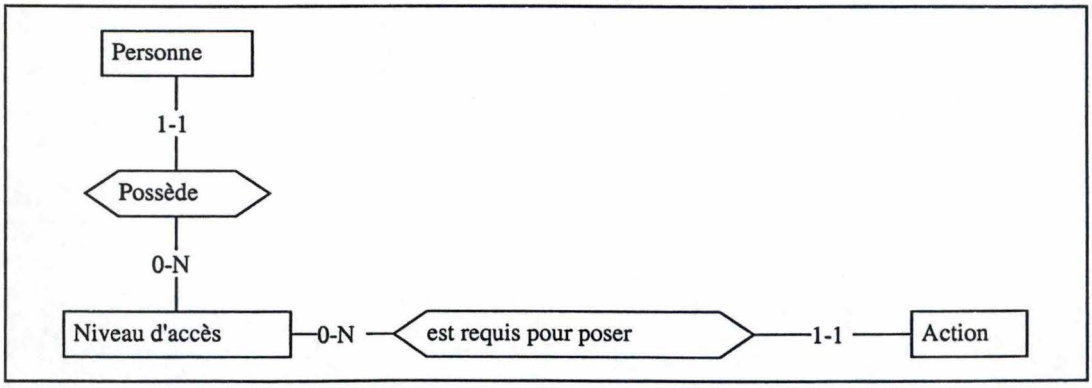

Figure 15 - Modèle des données - Facette "Sécurité"

8.4.7 Liste des Types d'Entité et leurs attributs

Remarque : les types d'associations ne sont pas repris comme attributs à ce stade-ci.

Personne
- Nom
- Prénom
- Adresse e-mail
- Fonction
- Coordinateur de sécurité
- Responsable hiérarchique

Accès
- Date d'effet
- Date d'expiration

Service
- Nom
- Description
- Responsable
- Gestionnaire d'accès
- Informations de configuration
- Méthode de suppression
- Méthode de contrôle de légitimité
- Scénario de test

Profil
- Nom
- Description

Groupe
- Nom
- Description

Demande d'accès
- Date d'introduction
- Date d'exécution
- Date d'effet demandée
- Date d'expiration demandée
- Jugée réalisable
- Jugée légitime

Type de demande d'accès

Demandeur
- Nom
- Prénom
- Adresse e-mail
- Fonction
- Coordinateur de sécurité
- Responsable hiérarchique

Système
- Nom
- Description

Bénéficiaire
- Nom
- Prénom
- Adresse e-mail
- Fonction
- Coordinateur de sécurité
- Responsable hiérarchique

Ressource
- Nom
- Description
- Gestionnaire
- Nature

Information de configuration
- Nom
- Libellé
- Description
- Catégorie
- Type
- Valeur par défaut
- Valeur minimale
- Valeur maximale

Règle de détermination

Chemin d'accès
- Nom
- Description
- Délai moyen d'intervention

Demande d'intervention
- Date d'exécution
- Opération

Autorité d'approbation
- Nom

Niveau d'accès
- Privilège

Action
- Numéro de fonctionnalité

8.5 Découpe fonctionnelle

8.5.1 Introduction

Dans le chapitre 4.6.2 'Modélisation des traitements', nous avons présenté le scénario d'une demande de création d'un nouvel accès. Les différentes actions de ce scénario ont alors été définies dans les grandes lignes. Nous allons dans ce chapitre traduire ces actions en fonctions et ensuite les regrouper en modules fonctionnels.

8.5.2 Méthode utilisée

Chaque action décrite dans le scénario générique est convertie en fonction que nous appelons « fonction métier ». On précise ensuite les caractéristiques propres à chaque fonction. On y ajoutera éventuellement des fonctions à caractère technique et utilitaire. Notons que les options prises lors des choix fondamentaux influencent le contenu de certaines fonctions. On indiquera enfin la liste des exigences remplies par ces fonctions, ce qui permettra d'assurer la traçabilité des étapes d'analyse.

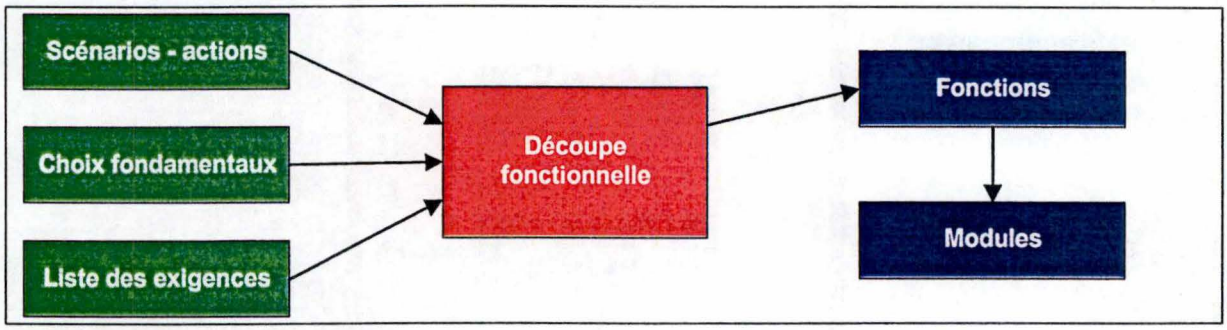

Figure 16 - Découpe fonctionnelle

Légende :
Les rectangles verts contiennent les informations nécessaires pour la découpe fonctionnelle
Le rectangle rouge symbolise la phase d'analyse
Les rectangles bleus indiquent les informations produites par l'analyse

8.5.3 Enchaînement des fonctions

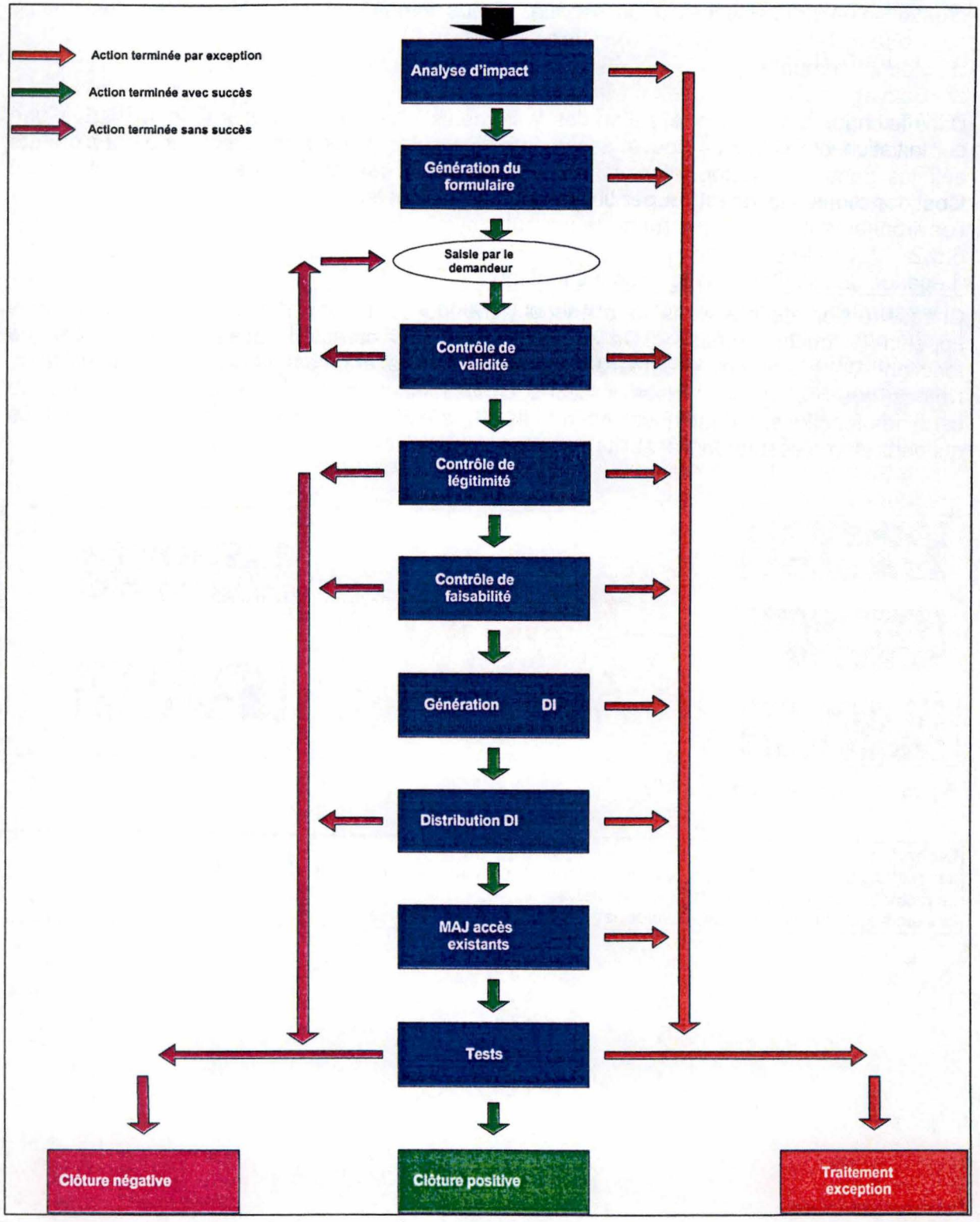

Figure 17 - Schéma d'enchaînement

Légende :
La flèche bleue indique le point d'entrée de l'enchaînement
L'ellipse blanche indique une action entreprise par le demandeur
Les rectangles en bleu indiquent les fonctions
La fin de l'enchaînement arrive à la fin de l'exécution du rectangle vert, rouge ou rose.
Ces rectangles comprennent également le retour d'information vers le demandeur et le bénéficiaire

8.5.4 Description des fonctions

Les fonctions décrites ci-dessous sont issues des actions du scénario de demande de création d'un nouvel accès à un service. Nous avons ajouté à celles-ci les fonctions à caractère technique et utilitaire; à savoir :

- Séquencement
- Sauvegarde
- Affichage
- Initiation WorkFkow

Ces fonctions assurent essentiellement la liaison entre les fonctions métiers et l'environnement du système (utilisateurs, autre système, ...)

Légende des libellés utilisés dans les tableaux.

Description : description sommaire de la fonction
Initié par : nom de la fonction ou de l'événement appelant
Pré-condition : état du système nécessaire pour l'accomplissement de la fonction
Post-condition : état du système après l'accomplissement de la fonction
Entrée : informations nécessaires pour l'accomplissement de la fonction
Sortie : informations générées par l'accomplissement de la fonction
Fonctionnement : description du principe de fonctionnement de la fonction
Exigence : liste des exigences satisfaites (chapitre 6.5) par l'exécution de la fonction

Séquencement	
Description	Cette fonction assure le séquencement (l'appel) des différentes fonctions correspondant à un enchainement donné.
Initié par	Dans le cas d'une demande d'accès, l'appel peut provenir soit : • Du gestionnaire des menus • D'une application externe • D'une fonction interne
Pré-condition	L'enchaînement à exécuter est défini
Post-condition	L'enchaînement a été exécuté dans son intégralité Ou Cette fonction se termine avec une exception
Entrée	Description de l'enchaînement à exécuter
Sortie	Statut de l'exécution de l'enchaînement sous forme booléenne (Vrai signifie succès, Faux signifie échec)
Fonctionnement	Cette fonction permet de lancer des fonctions en respectant un enchaînement donné.
Traçabilité	Pas d'application

Chaque type de demande d'accès correspond à un enchaînement particulier.
La table de décision ci-dessous constitue la matrice qui représente les fonctions lancées pour chaque type de demande d'accès. Ce tableau doit être considéré comme indicatif car il peut être adapté suivant le service demandé et le domaine d'application. L'opération de création (en rouge) a été décrite en détail par un scénario dans le chapitre précédent.

	Type de demande d'accès				
Fonctions à exécuter		**Suppression**	**Modification**	**Activation**	**Désactivation**
Saisie de la demande					
▪ Analyse d'impact	oui	oui	oui	Non	non
▪ Génération du formulaire	oui	non	oui	Non	non
Contrôles					
▪ Validité des données	oui	non	oui	non	non
▪ Légitimité	oui	oui*	oui	oui*	oui*
▪ Faisabilité	oui	non	oui	non	non
Intervention					
▪ Génération DI	oui	oui	oui	oui	oui
▪ Transmission DI	oui	oui	oui	oui	oui
Clôture					
▪ Mise à jour des accès	oui	oui	oui	oui	oui
▪ Test	non	non	non	oui	non
▪ Envoi confirmation	oui	oui	oui	oui	oui

(*) : Le choix positif s'explique par le fait que le contrôle de légitimité comporte deux volets : le contrôle vérifiant si le demandeur est habilité à effectuer une telle demande et le contrôle consistant à vérifier si on peut accorder au bénéficiaire le service demandé. Dans le cas présent, seul le contrôle de l'habilitation du demandeur sera effectué.

Sauvegarde	
Description	Cette fonction assure la persistance des informations.
Initié par	La fonction de mise à jour des accès existants
Pré-condition	Les informations présentes dans l'espace de stockage sont cohérentes
Post-condition	• Les informations fournies en entrée sont sauvegardées • Les informations présentes dans l'espace de stockage sont cohérentes
Entrée	Informations à sauvegarder
Sortie	Statut de l'exécution sous forme booléenne (Vrai signifie succès, Faux signifie échec)
Fonctionnement	Cette fonction sauvegarde physiquement l'information qui lui est passée.
Traçabilité	Contrainte non fonctionnelle n° 15

Affichage	
Description	Cette fonction assure le formatage et la présentation des informations à l'utilisateur du système.
Initié par	La fonction de séquencement
Pré-condition	-
Post-condition	Les informations fournies en entrée sont présentées à l'utilisateur
Entrée	• Informations à présenter • Modèles de présentation
Sortie	Statut de l'exécution sous forme booléenne (Vrai signifie succès, Faux signifie échec)
Fonctionnement	Cette fonction utilise des modèles qui, garnis par les données fournies par la fonction appelante, constituent les informations présentées à l'utilisateur.
Traçabilité	Contrainte non fonctionnelle n° 3 et 6

Initiation workflow	
Description	Cette fonction assure la communication entre le système et les différents acteurs via un système de WorkFlow.
Initié par	Dans le cadre d'une demande de création d'accès, l'appel de cette fonction peut provenir de : • La fonction de contrôle de légitimité • La fonction de contrôle de faisabilité • La fonction de distribution des DI • La fonction de test de bon fonctionnement de l'accès
Pré-condition	Scénario du workflow doit exister
Post-condition	Tous les acteurs présents dans le scénario ont rempli leur rôle respectif (compléter le formulaire)
Entrée	Scénario du workflow
Sortie	Statut de l'exécution sous forme booléenne (Vrai signifie succès, Faux signifie échec)
Fonctionnement	Cette fonction initie un workflow. Elle détermine un scénario de distribution de formulaires ainsi que la liste des destinataires. Cette fonction se termine lorsque le scénario s'est déroulé dans son entièreté, ce qui signifie dans la plupart des cas que l'ensemble des acteurs a complété et ensuite renvoyé le formulaire.
Traçabilité	Contrainte non fonctionnelle n° 7 et 14

Analyse d'impact	
Description	Cette fonction détermine les ressources impactées par une demande d'accès.
Initiée par	La fonction de séquencement
Pré-condition	La demande d'accès est introduite et le service demandé est connu.
Post-condition	Les ressources impactées sont déterminées ⇒ succès Ou Cette fonction se termine avec une exception
Entrée	• La demande d'accès • Liste complète des chemins d'accès existants utilisable pour le service demandé • Méthode de sélection des chemins d'accès
Sortie	Liste des ressources impactées sous la forme de chemins d'accès.
Fonctionnement	Le système SAGA propose trois alternatives pour la détermination des chemins d'accès. Celles-ci sont décrites en détail dans le chapitre 8.3.3.1 'Comment déterminer les CA's à prendre en compte pour mettre en œuvre une DA ? • Liste fixe de chemins d'accès Les chemins d'accès sont fixes, c'est à dire communs à toutes les demandes d'accès pour ce service. Ils sont déterminés lors de la mise en place du service. • Liste variable de chemins d'accès Les chemins d'accès peuvent varier en fonction de certains paramètres choisis (maximum 3) lors de la mise en place du service. Il s'agit donc de sélectionner m chemins d'accès parmi n possibles (m <= n). Le sélection est accomplie grâce à une requête SQL dynamique basée sur la valeur de ces paramètres . • Liste fournie par un module externe Si les deux solutions ci-dessus ne conviennent pas, il est possible d'appeler un module externe (plug-in) à qui on fournira les informations concernant la demande d'accès. Ce dernier renverra alors la liste des chemins d'accès impactés Le système utilisera la méthode de sélection définie lors de la mise en place du service.
Traçabilité	Exigence fonctionnelle n° 16

Génération du formulaire de demande d'accès (FDA)	
Description	Cette fonction consiste à générer le formulaire de demande d'accès qui sera présenté au demandeur.
Initié par	La fonction de séquencement
Pré-condition	Fonction d'analyse d'impact déroulée correctement
Post-condition	Le formulaire de demande d'accès est généré ⇒ succès ou Cette fonction se termine avec une exception
Entrée	La demande d'accès
Sortie	La demande d'accès avec le lien vers le formulaire de demande
Fonctionnement	Les ressources impactées sont connues car elles figurent dans les chemins d'accès qui ont été déterminés dans la fonction 'analyse d'impact'. Le gestionnaire de chaque ressource a indiqué lors de la mise en place du service les informations de configuration nécessaires pour la réalisation des interventions sur chaque ressource et cela pour chaque type de demande d'accès (création / suppression / modification / activation / désactivation). La fonction consolide ces informations de configuration, supprime les redondances et génère alors le formulaire.
Traçabilité	Contrainte non fonctionnelle n° 3, 6 et 14

Contrôle de validité	
Description	Cette fonction assure le contrôle de validité des informations saisies par le demandeur via le formulaire de demande d'accès.
Initié par	La fonction de séquencement (après l'action du demandeur qui consiste à remplir le formulaire de demande)
Pré-condition	• Formulaire de demande d'accès complété par le demandeur • Tous les types d'informations à contrôler possèdent un domaine de valeur dans le référentiel
Post-condition	Les informations saisies par le demandeur sont (correctes ⇒ succès ou incorrectes) ou Cette fonction se termine avec une exception
Entrée	• La demande d'accès comprenant le formulaire de demande complété par le demandeur • Le référentiel des informations de configuration
Sortie	• Statut du contrôle sous forme booléenne (Vrai signifie accepté, Faux signifie refusé) • Indication des zones erronées (si statut = Faux)
Fonctionnement	Chaque information contenue dans le formulaire possède un type. Le système SAGA gère un référentiel de ces informations. Dans ce dernier figure un domaine de valeur acceptable pour chaque type d'informations. Si une des valeurs saisies par le demandeur est en dehors du domaine, le formulaire est considéré comme incorrect. La fonction de séquencement le représente alors au demandeur en indiquant la ou les zone(s) erronée(s). Il s'agit donc ici d'une fonction itérative.
Traçabilité	Contrainte non fonctionnelle n° 3, 6 et 15

Contrôle de légitimité	
Description	Assure le contrôle de légitimité dans le scénario d'une demande d'accès
Initié par	La fonction de séquencement
Pré-condition	Les informations saisies par le demandeur dans le formulaire de demande d'accès sont correctes.
Post-condition	La demande est (légitime ⇒ succès ou illégitime) ou Cette fonction se termine avec une exception
Entrée	La demande d'accès
Sortie	• Statut du contrôle sous forme booléenne (Vrai signifie accepté, Faux signifie refusé) • Motif du refus (si statut = Faux)
Fonctionnement	Ce contrôle peut être effectué de différentes façons. Lors de l'ajout d'un nouveau service, on précise quel sera le comportement à adopter pour le contrôle de la légitimité d'une demande d'accès. Il existe trois comportements standards : • Pas de contrôle de légitimité Aucun contrôle n'est effectué, toutes les demandes sont alors considérées comme légitimes • Contrôle effectué par l'autorité d'approbation L'avis de l'autorité d'approbation est requis via à un mécanisme de workflow. • Contrôle effectué par un module externe Il est possible dans certains cas que le contrôle ne nécessite pas seulement l'approbation de personnes, mais qu'il soit basé sur un mécanisme de règles s'exécutant sur des informations spécifiques. Dans ce cas, un module externe, par exemple un script, permet d'effectuer le contrôle de légitimité. Nous supposons que toutes les informations nécessaires à ce contrôle sont fournies par la demande d'accès. Lorsque ce contrôle implique des personnes, le module externe doit pouvoir initier un workflow. Ce dernier recevra les informations nécessaires pour effectuer le traitement et renverra un statut sous forme booléenne. Consulter le chapitre 8.3.6 'Contrôle de légitimité' pour plus d'informations
Traçabilité	Exigence fonctionnelle n° 7 et 9 Contrainte non fonctionnelle n° 4, 6 et 10

Contrôle de faisabilité	
Description	Cette fonction assure le contrôle de faisabilité dans le scénario d'une demande d'accès.
Initié par	La fonction de séquencement
Pré-condition	les informations saisies par le demandeur dans le formulaire de demande d'accès sont correctes.
Post-condition	La demande est techniquement (réalisable ⇒ succès ou non) ou Cette fonction se termine avec une exception
Entrée	La demande d'accès
Sortie	Statut du contrôle sous forme booléenne (Vrai signifie accepté, Faux signifie refusé)
Fonctionnement	Cette fonction initie un workflow à destination du gestionnaire de chaque ressource impactée. Chacun de ces workflows se termine lorsque le gestionnaire de ressource indique s'il a approuvé ou non cette demande. Cette fonction est considérée comme terminée lorsque tous les workflows sont terminés.
Traçabilité	Exigence fonctionnelle n° 18 Contrainte non fonctionnelle n° 6 et 10

Génération des demandes d'intervention	
Description	Cette fonction traduit la demande d'accès en demandes d'intervention
Initié par	La fonction de séquencement
Pré-condition	Fonction précédente (dans le scénario) déroulée correctement.
Post-condition	Les DI ont été générées ⇒ succès ou Cette fonction se termine avec une exception
Entrée	La demande d'accès
Sortie	Autant de demandes d'intervention que de ressources impactées
Fonctionnement	Lors du remplissage du formulaire de demande d'accès, les informations de configuration nécessaires pour les interventions ont été récoltées. Il est question ici de générer un formulaire de demande d'intervention par ressource impactée comprenant les informations adéquates et qui sera transmis au gestionnaire de ressource correspondant.
Traçabilité	Exigence fonctionnelle n° 3 Contrainte non fonctionnelle n° 3, 6 et 14

Distribution et exécution des demandes d'intervention	
Description	Cette fonction assure la distribution des demandes d'intervention aux gestionnaires de ressource concernés
Initié par	La fonction de séquencement
Pré-condition	Fonction de génération des demandes d'intervention déroulée correctement.
Post-condition	((Les demandes d'intervention ont été transmises et (exécutées sur la ressource ⇒ succès ou non exécutées sur la ressource)) ou Cette fonction se termine avec une exception
Entrée	• Les demandes d'intervention à distribuer • Les références des gestionnaires des ressources impactées
Sortie	Le statut (réalisé ou pas) de chaque demande d'intervention est connu
Fonctionnement	Cette fonction initie un workflow pour chaque ressource impactée à destination de son gestionnaire. Chaque workflow se termine lorsque le gestionnaire de ressource indique s'il a ou non exécuté l'intervention sur la ressource. Cette fonction est considérée comme terminée lorsque tous les workflows sont terminés.
Traçabilité	Exigence fonctionnelle n° 3

Mise à jour des accès existants	
Description	Cette fonction consiste à apporter les modifications nécessaires pour que la liste des accès aux services reflète la réalité après que toutes les interventions aient été effectuées par les gestionnaires de ressource.
Initié par	La fonction de séquencement
Pré-condition	Fonction de distribution des demandes d'intervention déroulée correctement et toutes les DI ont été exécutées avec succès.
Post-condition	La liste des accès reflète la réalité ⇒ succès Ou Cette fonction se termine avec une exception
Entrée	La demande d'accès
Sortie	Statut de la mise à jour, sous forme booléenne (Vrai signifie succès, Faux signifie échec)
Fonctionnement	Cette fonction traduit la demande d'accès en accès. Elle fait ensuite appel à la fonction sauvegarde décrite précédement
Traçabilité	Contrainte non fonctionnelle n° 15

Tests de bon fonctionnement	
Description	Propose un ensemble de tests permettant de vérifier le bon fonctionnement de l'accès nouvellement créé.
Initié par	La fonction de séquencement
Pré-condition	Fonction précédente (dans le scénario) déroulée correctement et avec succès
Post-condition	Les tests entrepris sont (positifs ⇒ succès ou Cette fonction se termine avec une exception
Entrée	La demande d'accès
Sortie	Statut des tests sous forme booléenne (Vrai signifie tests positifs, Faux signifie tests négatifs)
Fonctionnement	Les tests peuvent se dérouler de deux manières différentes: • Automatique Un module externe est chargé de tester l'accès sans l'aide d'une personne. • Avec intervention humaine Un workflow est initié avec comme acteur le bénéficiaire et les gestionnaires de ressource concernés. Un formulaire décrivant les tests à exécuter est envoyé au bénéficiaire. Celui réalise ces tests et complète le formulaire avec les résultats. En cas de test négatif, le formulaire est redirigé vers les gestionnaires de ressources concernés qui entreprennent les actions de correction nécessaires. En cas de problème (terminaison avec exception), un traitement manuel est requis.
Traçabilité	Exigence fonctionnelle n° 20 Contrainte non fonctionnelle n° 2 et 6

Retour d'information	
Description	Cette fonction consiste à informer le demandeur et le bénéficiaire du statut final de la demande d'accès.
Initié par	La fonction de séquencement
Pré-condition	
Post-condition	Le demandeur et le bénéficiaire sont informés ⇒ succès Ou Cette fonction se termine avec une exception
Entrée	Le statut final de la dernière fonction qui a été exécutée sur la demande d'accès
Sortie	Le message : • L' accès est disponible • L'accès est indisponible pour cause d'un refus dans un contrôle • L'accès est indisponible pour cause d'erreur du système (exception dans une fonction précédente)
Fonctionnement	Cette fonction transmet un message standard au demandeur et au bénéficiaire. Ce message comprend essentiellement les références de la demande ainsi que dans certains cas une notice d'utilisation (uniquement pour le bénéficiaire) si l'accès demandé est disponible.
Traçabilité	Exigence fonctionnelle n° 2 Contrainte non fonctionnelle n° 6, 11 et 14

8.5.5 Classification des modules fonctionnels

On peut distinguer dans le système trois catégories de modules possédant chacun des types de fonctionnalités différentes :

- *Les modules de gestion*
  Ceux-ci assurent les fonctions « métiers » du système, notamment la gestion des demandes d'accès.

- *Les modules de configuration*
  Assure les fonctions de configuration du système, comprenant par exemple la gestion des ressources et des chemins d'accès, des services.

- *Les modules techniques et utilitaires*
  Fournit les fonctions utilitaires utilisées par les modules des catégories ci-dessus. Ces fonctions comprennent principalement la communication entre le système et le monde extérieur, la persistance des informations, l'exécution des scénarios.

8.5.6 Regroupement en modules des fonctions

Nous allons placer les fonctions dans des modules fonctionnels. Chaque module fonctionnel devrait correspondre à une classe d'objet dont les fonctions en seraient les méthodes. Ci-dessous ne figure que les modules impliqués dans le scénario décrit dans le chapitre 8.2.4 'Scénario générique d'une demande d'accès'. La traduction précise de l'ensemble des modules en classe d'objets figurera dans le chapitre suivant.

Nom du module	**Type**	**Fonctions**
Séquenceur	Technique-utilitaire	• Séquencement
Persistance	Technique-utilitaire	• Sauvegarde
Présentation	Technique-utilitaire	• Affichage
Workflow	Technique-utilitaire	• Initiation workflow
Traitement DA (demande d'accès)	Gestion	• Analyse d'impact • Génération du formulaire • Contrôle de validité des données • Contrôle de légitimité • Contrôle de faisabilité • Génération des demandes d'intervention • Distribution des demandes d'intervention • Mise à jour des accès • Tests • Retour d'informations

9 Conception du système

9.1 Introduction

La phase de conception du système consiste en différentes étapes qui permettent de traduire les modules fonctionnels et le modèle des données en classes d'objet. Lors de cette phase, nous déterminons également l'architecture cible du futur logiciel.
Notons que pour la conception du système, nous avons travaillé en profondeur en nous focalisant sur le seul scénario de traitement d'une demande de création d'un accès.

9.2 Architecture

9.2.1 Introduction

Nous avons défini dans des chapitres précédents la liste des modules fonctionnels ainsi que les réponses à des questions qui nous semblaient fondamentales pour l'implémentation du système SAGA. Toutes ces informations vont nous être utiles pour déterminer la forme définitive de SAGA en terme d'architecture. Ce chapitre définit les lignes directrices pour l'implémentation du système. Vous trouverez ci-dessous les choix architecturaux ainsi que l'argumentation qui a mené à ces choix.

9.2.2 Méthode

Nous allons déterminer dans ce chapitre le modèle utilisé pour l'implémentation du système SAGA. Le choix du modèle tient compte des contraintes non fonctionnelles et des indications fournies dans le chapitre 8.3 'Discussion des choix fondamentaux'. La distribution des modules fonctionnels dans les couches applicatives est réalisée suivant les catégories(voir 8.5.5. 'Classification des modules fonctionnels' de ces modules.

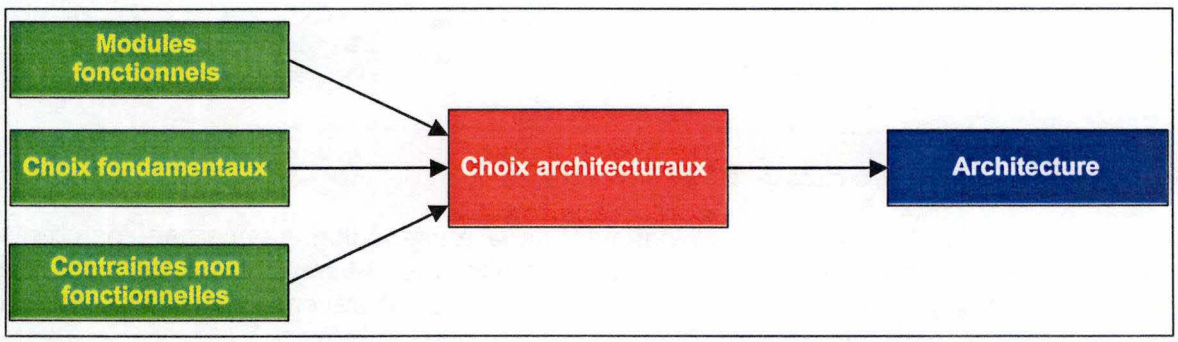

Figure 18 - Choix architecturaux

9.2.3 Choix du modèle

Le choix du modèle a été fixé en tenant compte des exigences suivantes :

- La facilité d'intégration dans le système informatique existant de la société
- La sécurité, c'est à dire le contrôle d'accès aux fonctionnalités et aux informations
- La cohérence des informations traitées par le système
- La performance du système en terme de rapidité, de disponibilité
- L'accessibilité de l'application
- L'extensibilité du système

Nous avons choisi, dans le cadre de ce projet une architecture qui suit le modèle 3-tiers de type client léger.

Le schéma ci-dessous représente les principes de base d'une telle architecture.

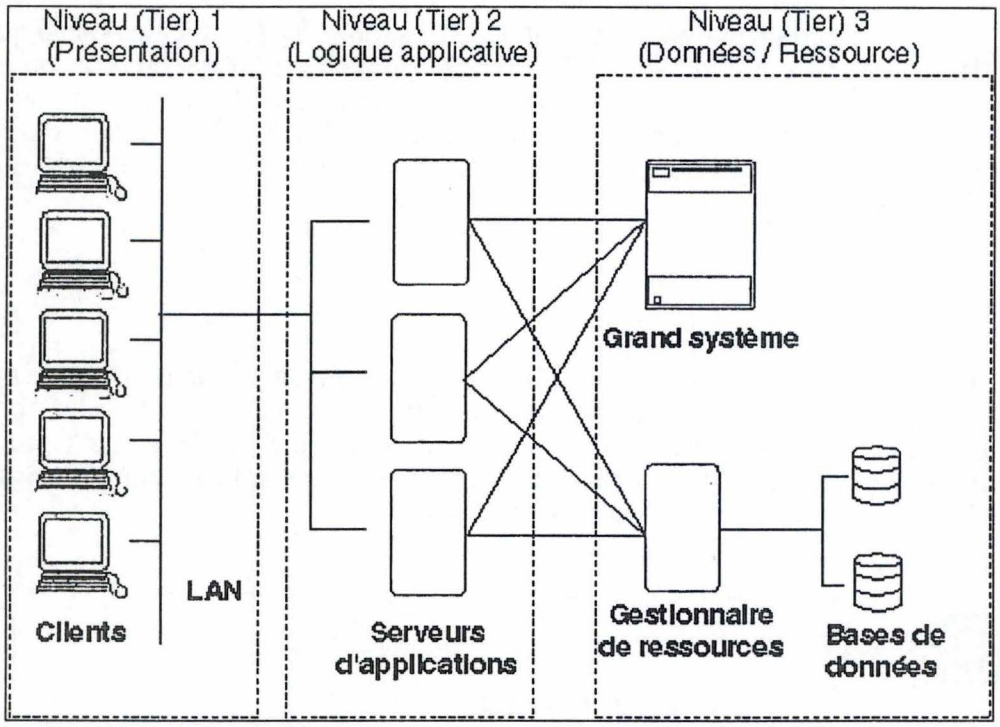

Figure 19 - Exemple d'architecture 3-tiers (source IBM)

Le client léger (navigateur) s'occupe, uniquement de la présentation. Concernant les clients légers, ils s'intègrent selon deux modes dans l'architecture 3-tiers : le mode passif qui consiste à la lecture simple d'éléments HTML et le mode actif à travers le téléchargement et l'exécution de micro applications, ActiveX ou Applets Java selon l'offre.

Dans ce type architecture, une partie (que l'on nommera persistance) gère l'accès aux données. Cette couche assure également la cohérence des informations stockées.

Entre ces deux parties, est placée une couche modulaire qui traite la logique applicative (les traitements métier). Cette couche est structurée en briques logicielles indépendantes, réutilisables, communicantes et pouvant être réparties sur une ou plusieurs machines. On parle de " middleware " ou serveur d'application pour désigner cette partie intermédiaire.

Conclusions
L'utilisation de ce modèle correspond bien à nos besoins car, en étant modulaire, il permet une intégration plus facile avec l'existant que l'utilisation d'une application monolithique.

D'autre part, pouvoir dissocier la couche persistance des autres couches peut garantir une sécurité accrue en terme de cohérence des informations. Ce type de modèle permet de limiter les clients communicants avec la couche persistance à la seule couche traitement métier, ce qui améliore également la sécurité globale du système.

La modularité de cette architecture permet d'assurer une bonne évolutivité du système et, en cas de problème (par exemple de performance), de corriger ou renforcer uniquement la couche ou le module qui a des faiblesses.

L'utilisation de clients légers du type navigateur offre des possibilités très étendues en terme d'accessibilité et de déploiement.

9.2.4 Détail de la couche présentation

Notre système doit communiquer avec le monde extérieur notamment pour recevoir les demandes d'accès et pour transmettre les demandes d'interventions sur les ressources. Les acteurs concernés par cette communication sont :

- *Les utilisateurs de SAGA*
  Ces utilisateurs sont en fait ceux qui utilisent interactivement l'interface de SAGA pour lancer des actions ou récolter de l'information. Dans ce cas de figure, il s'agit non seulement de faire communiquer les utilisateurs et notre système, mais aussi de présenter de manière simple et conviviale les informations aux utilisateurs.

- *Des systèmes externes*
  La communication entre SAGA et ces systèmes externes peut être initiée soit par SAGA (SAGA jouant le rôle de client) , soit par les systèmes externes (SAGA jouant le rôle du serveur).

  Exemples de cas ou SAGA joue le rôle du serveur :
  - Une application de gestion des ressources humaines peut signaler à SAGA le départ d'un collaborateur.
  - Une application externe souhaite interroger SAGA sur les accès impactant une ressource particulière

  Nous pouvons déterminer ici l'interface et le moyen de transport de l'information que devront adopter ces systèmes externes pour communiquer avec SAGA. L'utilisation des mêmes moyens de transport que pour les utilisateurs permet de simplifier le contenu de la couche Présentation. Il est important également de ne pas s'éloigner des standards actuels.

Exemples de cas ou SAGA joue le rôle du client :

- Connexion à un programme de workflow existant pour transmettre les demandes d'interventions.
- Envoi à une application de supervision une alerte de mauvais fonctionnement de notre système.
- Envoi d'un courrier électronique à un bénéficiaire pour lui signaler que son accès est disponible.

Étant donné qu'il n'existe pas (encore) de standard permettant la communication avec ces applications, nous devrons écrire un client spécifique pour chacune de ces applications.

Nous allons placer dans la couche applicative, les modules fonctionnels chargés d'assurer l'interface avec les utilisateurs et les systèmes externes. Le module de présentation sera également placé à ce niveau.

9.2.5 Détail de la couche persistance

La persistance des informations dans le système SAGA est assurée par trois canaux séparés :

- *Via un système d'annuaire*
  On stocke dans cet annuaire des objets dont la mouvance est faible, c'est à dire que l'on aura peu d'actions d'écriture et de modifications, par contre l'accès en lecture est optimisé. De ce fait, nous pouvons stocker dans cet annuaire les objets tels que les personnes, groupes, profils, et des ressources.
  Les systèmes d'annuaires permettent également de déléguer éventuellement la gestion de ces objets à d'autres applications ou à des personnes, par exemple, on pourrait envisager de déléguer la gestion des personnes et des groupes au département des ressources humaines.

- *Via un système de base de données relationnelles (SGBDR)*
  Nous stockons dans une base de données relationnelles les objets tels que la liste des accès existants, des chemins d'accès,... qui nécessitent une mise à jour fréquente et qui sont l'objet de requêtes complexes.

- *Sous la forme de fichiers plats structurés*
  L'avantage des fichiers plats est la simplicité d'édition et de consultation. Les principales informations qui seront stockées sous cette forme seront les demandes d'accès, les demandes d'intervention et les fichiers de configuration.

Nous allons placer dans cette couche le module fonctionnel persistance qui aura pour objectif de déterminer l'espace de stockage parmi les trois définis ci-dessus et d'assurer l'aspect transactionnel des opérations d'écriture.

9.2.6 Détail de la couche logique applicative

La couche application assure les fonctions de traitement 'métier' de l'application.
Dans notre cas, on distingue les composants (briques logicielles) génériques qui sont communes à tous les domaines d'application et les composants spécifiques à certains domaines. Par exemple, il sera ainsi parfois nécessaire d'adapter les composants traitant notamment les différents contrôles (légitimité, faisabilité).

Nous trouverons dans cette couche également certains modules qualifiés de techniques-utilitaires chargés d'assurer des fonctions telles que la sécurité (contrôle des accès aux fonctionnalités du système) ou l'ordonnancement de tâches.

9.2.7 Schéma global de l'architecture de SAGA

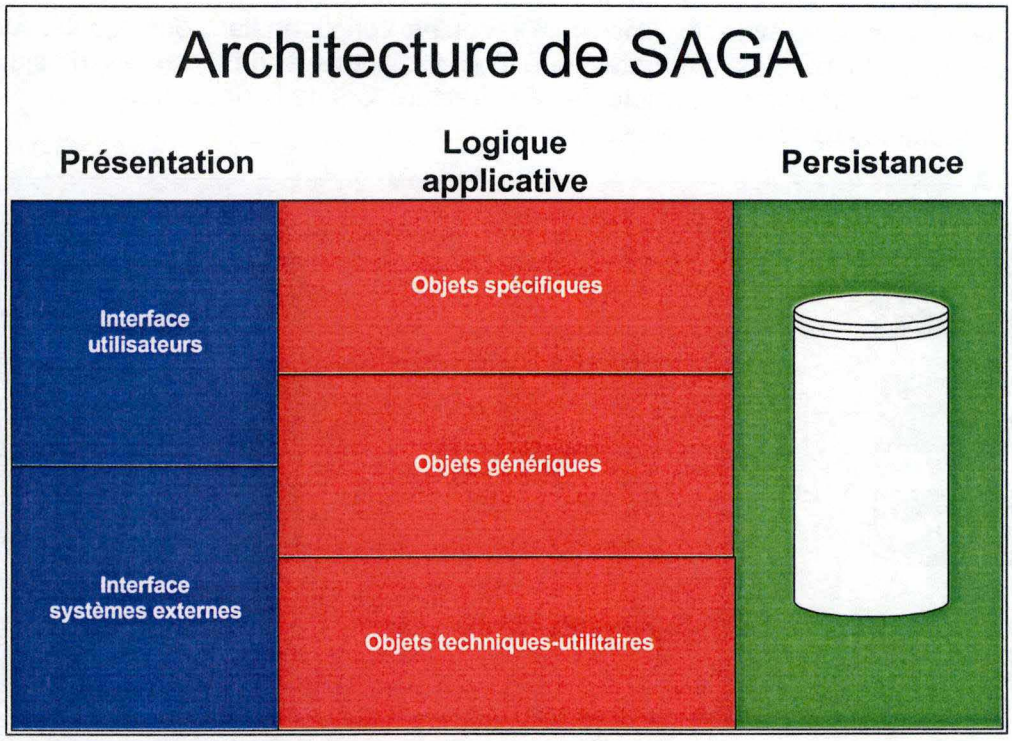

Figure 20 - Architecture de SAGA

9.3 Découpe en classes d'objets

9.3.1 Introduction

Dans ce chapitre, nous allons à partir du modèle des données, de la découpe en modules fonctionnels ainsi que des choix architecturaux décrits dans les chapitres précédents, pouvoir définir ici la liste des différentes classes d'objets de notre système. Nous complèterons chacune de ces classes par les méthodes et attributs correspondants.

9.3.2 Méthode utilisée

En se basant sur le modèle des données, on déterminera les classes d'objets de base de notre système ; chaque type d'entité devient une classe d'objet.
Ensuite nous passerons en revue la liste des modules fonctionnels pour ajouter de nouvelles classes d'objets ou des méthodes que l'on assignera à des classes d'objets existantes. Nous tiendrons également compte de l'architecture lors de la découpe.
Nous établirons ensuite les relations entre ces objets.

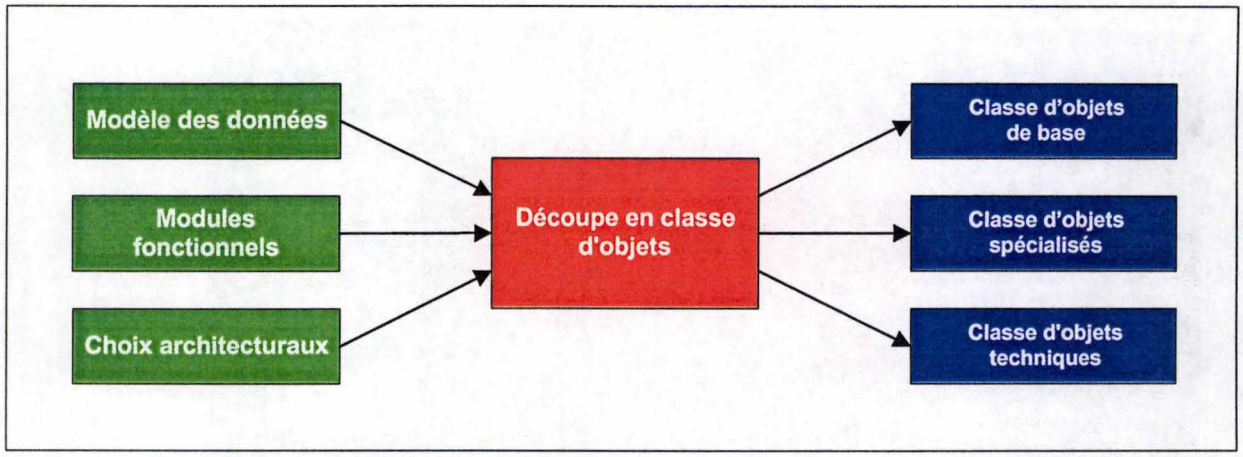

Figure 21 - Découpe en classe d'objets

9.3.3 Tableau des classes d'objets

Le tableau ci-dessous reprend l'ensemble des classes d'objets extraites suivant la méthode explicitée au point 9.3.2. Ce tableau n'est pas exhaustif, du fait que l'on décrit dans ce document que le scénario générique d'une demande d'accès de type création.

Nom	Type	Origine
Personne	Base	Modèle des données
Service	Base	Modèle des données
Ressource	Base	Modèle des données
Chemin d'accès (CA)	Base	Modèle des données
Groupe de personnes	Base	Modèle des données
Profil	Base	Modèle des données
Accès existant	Base	Modèle des données
Demande d'accès (DA)	Base	Modèle des données Découpe fonctionnelle
Demande d'intervention (DI)	Base	Modèle des données Découpe fonctionnelle
Formulaire de demande d'accès (FDA)	Base	Modèle des données Découpe fonctionnelle
Présentation	Technique	Choix architecturaux
Dispatcher	Technique	Choix architecturaux
Interface application externe	Technique	Choix architecturaux
Séquenceur	Technique	Modules fonctionnels
Interface workflow	Technique	Modules fonctionnels
Interface Scheduler	Technique	Choix architecturaux
Configuration système	Technique	Choix architecturaux
Consultation	Technique	Choix architecturaux
Information de configuration	Base	Modèle des données
Sécurité	Technique	Choix architecturaux
Persistance	Technique	Modules fonctionnels
Auditing & Alarming	Technique	Choix architecturaux
Reporting	Technique	Choix architecturaux
Légitimité	Base	Modèle des données Modules fonctionnels
Faisabilité	Base	Modèle des données Modules fonctionnels

9.3.4 Liste des classes d'objets

Vous trouverez ci-dessous la légende des libellés utilisés dans les tableaux

Fonction : Rôle de la classe d'objet
Description : Brève description de la classe d'objets
Type : Générique/spécifique (lié à un domaine d'application)
Interface : Liste des méthodes exposées par cette classe d'objet
Attribut : Liste des informations qualifiant la classe d'objet
Persistance : Méthode utilisée pour assurer la persistance de cette classe d'objet
Remarque : Remarque particulière concernant la classe d'objet

Formalisme utilisé pour les méthodes :
Nom_méthode(argument(s):Type):valeur_retour :Type
Formalisme utilisé pour les attributs : Nom_attribut :Type

Nous considérons comme implicite les méthodes permettant de mettre à jour la valeur des attributs ainsi que leur affichage. Elles ne figurent donc pas dans les tableaux ci-dessus.

Personne	
Fonction	Qualifie une personne et en assure la gestion
Type	Générique
Interface	New() → Statut :Integer ▪ Crée une nouvelle instance de la classe personne
	Load(Référence : Integer)→ Statut :Integer ▪ Charge une instance d'une personne existante
	Save()→ Statut :Integer ▪ Sauvegarde les informations d'une personne
	Delete()→ Statut :Integer ▪ Supprime physiquement les informations de la personne
	AddProfile(Nom Profil :Profil) → Statut :Integer ▪ Associe un profil à la personne
	RemoveProfile(Nom Profil :Profil) → Statut :Integer ▪ Retire le profil de la personne
	Display() → Statut :Integer ▪ Affiche les attributs de la personne
Attributs	Référence : Integer ▪ Identifiant de la personne
	Nom : String ▪ Nom de la personne
	Prénom : String ▪ Prénom de la personne
	Commentaire : String ▪ Information optionnelle
	Adresse Email : String ▪ Adresse Email
	Nom Profil : Profil ▪ Profil de la personne
	Responsable hiérarchique : Personne ▪ Responsable hiérarchique de la personne
	Coordinateur décentralisé : Personne ▪ Coordinateur décentralisé de la personne
	Accès existants[0-n] : Accès existant ▪ Liste des accès existants de la personne
Persistance	Les informations qualifiant une personne seront sauvegardées dans un annuaire.

Service	
Fonction	Qualifie un service et en assure la gestion
Type	Générique
Interface	New()➔ Statut :Integer ▪ Crée un nouveau service
	Load(Référence : Integer)➔ Statut :Integer ▪ Charge un service existant
	Save()➔ Statut :Integer ▪ Sauvegarde le service
	Delete() ➔ Statut :Integer ▪ Supprime le service
	AddCA(CA :Chemin d'accès) ➔ Statut :Integer ▪ Ajoute un chemin d'accès au service
	RemoveCA(CA :Chemin d'accès) ➔ Statut :Integer ▪ Retire un chemin d'accès du service
	DisplayCA() ➔ Liste[0-n] : CA ▪ Affiche la liste des CA utilisés pour ce service
	CtrlDisponibility(Accès : Accès existant) ➔ Statut :Integer ▪ Contrôle la disponibilité de l'accès à ce service
	SearchAccessManager(Bénéficiaire : Personne)➔ Statut :Integer ▪ Détermine le gestionnaire d'accès à ce service pour le bénéficiaire indiqué
Attributs	Référence : Integer ▪ Référence(identifiant) du service
	Nom : String ▪ Nom du service
	Description : String ▪ Description du service
	Responsable : Personne ▪ Nom du responsable de ce service
	Gestionnaire d'accès : Personne ▪ Nom du gestionnaire d'accès de ce service pour un bénéficiaire donné
	Liste des CA[1-n] : CA ▪ Liste des chemins d'accès
	Liste d'accès : Personne/Groupe de personne ▪ Liste des bénéficiaires ou de groupes de bénéficiaires pouvant utiliser ce service
	Liste d'informations de configuration[0-n,5] : Information de configuration ▪ Liste des informations de configuration liée au service pour chaque type de demande d'accès.
	Méthode de suppression : Integer ▪ Méthode adoptée pour la suppression d'un accès existant à ce service
	Méthode de contrôle de légitimité : Integer ▪ Méthode adoptée pour le contrôle de légitimité d'une demande d'accès existant à ce service.
	Scénario de test : Integer ▪ Description du scénario de test utilisé pour vérifier le bon fonctionnement d'un accès existant à ce service.
Persistance	Les informations qualifiant un service seront sauvegardées dans un système de base de données relationnelles et sous la forme de fichiers plats de configuration.

Ressource	
Fonction	Qualifie une ressource et en assure la gestion
Type	Générique
Interface	New() → Statut :Integer ▪ Crée une nouvelle ressource
	Load(Référence : Integer) → Statut :Integer ▪ Charge les informations qualifiant une ressource existante
	Save() → Statut :Integer ▪ Sauvegarde les informations qualifiant une ressource existante
	Delete()→ Statut :Integer ▪ Suppression des informations qualifiant une ressource existante
Attributs	Référence : Integer ▪ Référence(Identifiant) de la ressource
	Nom : String ▪ Nom de la ressource
	Description : String ▪ Description de la ressource
	Gestionnaire : Personne ▪ Gestionnaire de cette ressource
	Nature : String ▪ Nature de cette ressource (firewall/porte blindée/barrière amovible/…)
	Liste des informations de configuration [0-n][5][0-m]: Information de configuration ▪ Liste des informations de configuration nécessaires pour l'intervention sur la ressource. Il faut noter que n représente le nombre maximum de services disponibles, 5 le nombre de type de demandes d'accès à ces services (Création/Suppression/Modification/Activation/désactivation) et m le nombre d'informations requises pour chaque type de demandes.
Persistance	Les informations qualifiant une ressource seront sauvegardées dans un système de base de données relationnelles.

Chemin d'accès (CA)	
Fonction	Qualifie un chemin d'accès d' accès et en assure la gestion
Type	Générique
Interface	New()➔ Statut :Integer ▪ Crée un nouveau CA
	Load(Référence : Integer) ➔ Statut :Integer ▪ Charge les informations qualifiant un CA existant
	Save()➔ Statut :Integer ▪ Sauvegarde les informations qualifiant un CA existant
	Delete()➔ Statut :Integer ▪ Supprime le CA
	AddRessource(Nom_ressource :Ressource, Position :Integer) ➔ Statut :Integer ▪ Ajoute une ressource à ce CA
	RemoveRessource(Nom_ressource :Ressource) ➔ Statut :Integer ▪ Supprime une ressource de ce CA
Attributs	Référence : Integer ▪ Identifiant du chemin d'accès
	Nom : String ▪ Nom du chemin d'accès
	Description : String ▪ Description du chemin d'accès
	Liste des ressources [1-n]: Ressource ▪ Liste ordonnée des ressources affectées à ce chemin d'accès
	Nom du service : Service ▪ Service qui utilise ce CA
Persistance	Les informations qualifiant un chemin d'accès seront sauvegardées dans un système de base de données relationnelles.

Groupe de personnes	
Fonction	Qualifie un groupe de personnes et en assure la gestion
Type	Générique
Interface	New()➔ Statut :Integer ▪ Crée un nouveau groupe
	Load(Référence : Integer) ➔ Statut : Integer ▪ Charge un groupe existant
	Save() ➔ Statut : Integer ▪ Sauvegarde le groupe
	Delete() ➔ Statut : Integer ▪ Supprime le groupe
	AddPerson(person) ➔ Statut : Integer ▪ Ajoute une personne dans le groupe
	RemovePerson(person) ➔ Statut : Integer ▪ Retire une personne du groupe
	AddProfil() ➔ Statut : Integer ▪ Associe un profil à un groupe
	RemoveProfil() ➔ Statut : Integer ▪ Retire un profil à un groupe
Attributs	Référence : Integer ▪ Identifiant du groupe
	Nom : String ▪ Nom du groupe
	Description : String ▪ Description du groupe
	Liste de personnes [0-n] : Personne ▪ Liste des personnes composant le groupe
	Accès existants[0-n] : Accès existant ▪ Liste des accès existants du groupe
Persistance	Les informations qualifiant un groupe de personnes seront sauvegardées dans un annuaire.

Profil	
Fonction	Qualifie un profil et en assure la gestion
Type	Générique
Interface	New()➔ Statut :Integer ▪ Crée un nouveau profil
	Load(Référence : Integer)➔ Statut :Integer ▪ Charge un profil existant
	Save() ➔ Statut :Integer ▪ Sauvegarde le profil
	Delete() ➔ Statut :Integer ▪ Supprime le profil
	AddService(Nom_service : Service) ➔ Statut :Integer ▪ Ajoute un service à un profil
	RemoveService(Nom_service :Service) ➔ Statut :Integer ▪ Retire un service d'un profil
Attributs	Référence : Integer ▪ Référence(Identifiant) du profil
	Nom : String ▪ Nom de ce profil
	Description : String ▪ Description de ce profil
	Liste des services[1-n] : Service ▪ Liste des services liés à ce profil
Persistance	Les informations qualifiant un profil seront sauvegardées dans un annuaire.

Accès existant	
Fonction	Qualifie un accès existant et en assure la gestion
Type	Générique
Interface	New()➔ Statut :Integer ▪ Crée un nouveau accès (Attention, on parle ici de la mémorisation d'un accès résultant d'une demande de création d'un accès)
	Load(Référence : Integer)➔ Statut :Integer ▪ Charge les informations qualifiant l'accès
	Save() ➔ Statut :Integer ▪ Sauvegarde les informations qualifiant l'accès
	Delete() ➔ Statut :Integer ▪ Supprime les informations qualifiant l'accès
Attributs	Référence : Integer ▪ Identifiant de l'accès
	Date d'effet : Date ▪ Date à laquelle l'accès a été opérationnel
	Date d'expiration : Date ▪ Date à laquelle l'accès devra être supprimé ou désactivé
	Demande d'accès : DA ▪ Référence de la demande d'accès
Persistance	Les informations qualifiant un accès existant seront sauvegardées dans un système de base de données relationnelles.

Demande d'accès (DA)	
Fonction	Qualifie une demande d'accès
Type	Générique
Interface	New(TypeDA : String)➔ Statut :Integer ▪ Crée une nouvelle demande d'accès du type passé en paramètre ▪ Type de demande d'accès = (création|suppression|modification|activation|désactivation)
	Load(Référence : Integer) ➔ Statut :Integer ▪ Charge une demande d'accès existante
	Save()➔ Statut :Integer ▪ Sauvegarde une demande d'accès
	Delete()➔ Statut :Integer ▪ Supprime une demande d'accès
	SearchCA()➔ Statut :Integer ▪ Recherche la ou les listes des chemins d'accès lié au service requis.
	MakeFDA()➔ Statut :Integer ▪ Génère un formulaire vierge de demande d'accès (FDA)
	CtrlValidity()➔ Statut :Integer ▪ Supprime une demande d'accès
	CtrlLegitimity() ➔ Statut :Integer ▪ Instancie la classe d'objet chargée d'effectuer le contrôle de légitimité
	CtrlFaisability() ➔ Statut :Integer ▪ Instancie la classe d'objet chargée d'effectuer le contrôle de faisabilité
	MakeDI() ➔ Statut :Integer ▪ Génère les demandes d'interventions
	SaveAccess() ➔ Statut :Integer ▪ Sauvegarde les accès résultants de la demande d'accès (DA)
	TestAccess() ➔ Statut :Integer ▪ Teste le fonctionnement des accès résultant d'une demande.
Attributs	Référence : Integer ▪ Référence (identifiant)de la DA
	Demandeur : Personne ▪ Référence du demandeur
	Bénéficiaire : Personne ▪ Référence du bénéficiaire
	Service demandé : Service ▪ Référence du service souhaité
	Date de demande : Date ▪ Date de la demande
	Date d'exécution : Date ▪ Date souhaitée de l'exécution
	Date d'effet : Date ▪ Date souhaitée de la prise d'effet de l'accès
	Date d'expiration : Date ▪ Date souhaitée de la fin de l'accès
	Formulaire de demande d'accès : FDA ▪ Référence du formulaire de demande d'informations
	Liste des CAs [1-n] : CA ▪ Liste des chemins d'accès
Persistance	Liste des DIs [1-n] : DI ▪ Liste des demandes d'intervention
	Les informations qualifiant une demande d'accès seront sauvegardées dans un système de base de données relationnelles.

Demande d'intervention (DI)	
Fonction	Qualifie une demande d'intervention
Description	Les demandes d'interventions sont générées dans le scénario d'une demande d'accès.
Type	Générique
Interface	New() → Statut :Integer ▪ Crée une nouvelle DI
	Load(Référence : Integer)→ Statut :Integer ▪ Charge les informations d'une DI existante
	Save()→ Statut :Integer ▪ Sauvegarde les informations d'une DI
	Delete()→ Statut :Integer ▪ Suppression des informations concernant une DI
	SetOperation(Paramètres :Liste[0-n] d'informations de configuration) → Statut :Integer ▪ Fixe l'opération souhaitée ainsi que les paramètres
	DeleteOperation() → Statut :Integer ▪ Suppression de l'opération
	SendDI() → Statut :Integer ▪ Distribution de la demande d'intervention (DI)
Attributs	Référence : Integer ▪ Référence(Identifiant) de la DI
	Demande d'accès : DA ▪ Référence (Identifiant) de la DA
	Gestionnaire de ressource : Personne ▪ Nom (identifiant) du gestionnaire de la ressource
	Date d'exécution : Date ▪ Date d'exécution souhaitée de l'intervention
	Opération souhaitée : Opération ▪ Code opération souhaitée sur la ressource
	Paramètres de l'opération [0-n] : Informations de configuration ▪ Informations nécessaires pour l'opération
Persistance	Chaque demande d'intervention sera sauvegardée sous la forme d'un fichier au format XML.

Formulaire de demande d'accès (FDA)	
Fonction	Qualifie et assure la gestion d'un FDA.
Description	Le FDA , une fois complété par le demandeur, contiendra toutes les informations nécessaires à la paramétrisation des ressources impactées par une demande d'accès.
Type	Générique
Interface	New() → Statut :Integer ▪ Crée un formulaire vide
	Load(Référence : Integer) →: Statut :Integer ▪ Charge un formulaire existant
	Save() → Statut :Integer ▪ Sauvegarde le formulaire
	Delete() → Statut :Integer ▪ Supprime le formulaire
	Generate() → Statut : Integer ▪ Ajoute au formulaire la liste des informations qui seront réclamées au demandeur
Attributs	Référence : Integer ▪ Référence (identifiant) du formulaire
	Date de création : Date ▪ Date de création du formulaire
	Demande d'accès : DA ▪ Référence de la demande d'accès
	Liste d'informations demandées [0-n] : Informations de configuration ▪ Liste des informations de configuration
Persistance	Le formulaire de demande d'accès sera sauvegardé sous la forme d'un fichier au format XML.

Présentation	
Fonction	Assure l'interface entre SAGA et les utilisateurs du système
Description	Cette classe d'objet est chargée de présenter : • Les différents menus • Les formulaires • Les résultats des requêtes Les pages d'informations renvoyées aux utilisateurs sont en fait des modèles garnis par les informations fournies par les fonctions appelantes
Type	Générique
Interface	DisplayMenu(Référence : Integer) → Statut :Integer ▪ Affiche un menu
	DisplayRequestForm(Référence : Integer) → Statut :Integer ▪ Affiche un formulaire
	DisplayResults(Référence : Integer) → Statut :Integer ▪ Affiche les résultats d'une requête
Attributs	Menus [1-n] : Object ▪ Liste des menus
	Formulaires [1-n] : Object ▪ Liste des formulaires
	Modèles : Fichiers XML ▪ Modèles des pages présentées aux utilisateurs
Persistance	Les modèles sont enregistrés sous formes de fichiers XML

Dispatcher	
Fonction	Assure l'appel des différentes fonctionnalités de SAGA.
Description	Cette classe d'objet a pour objectif de lancer les différentes actions requises par les utilisateurs ou les applications externes. A cette fin , il garde la trace de toutes les sessions en cours
Type	Générique
Interface	Dispatch(Session : Objet, Fonction : Objet) → Statut :Integer ▪ Distribue les actions
Attributs	Sessions [0-n] : Objet ▪ Liste des sessions en cours

Interface application externe	
Fonction	Assure la communication entre les applications externes et SAGA.
Description	La fonction de cet objet est de fournir à des applications externes la possibilité d'exécuter des méthodes incluses dans SAGA.
Type	Spécifique
Interface	ExecuteFonction (Fonction : Object) → Retour : Object ▪ Exécute la fonction indiquée

Séquenceur	
Fonction	Exécute séquentiellement les actions reprises dans un scénario
Type	Générique
Interface	New(Scénario : Object) → Référence : Integer ▪ Exécute les actions reprises dans le scénario
	Pause(Référence : Integer) → Statut : Integer ▪ Interrompt l'exécution du scénario
	ReStart(Référence : Integer) → Statut : Integer ▪ Redémarre un scénario qui a été interrompu
	GetStatus(Référence : Integer)→ Statut : Integer ▪ Donne le statut d'un scénario en cours de traitement
Attributs	Scénario : Objet ▪ Nom du scénario qui est interprété
Persistance	Les informations qualifiant cette classe d'objet seront sauvegardées dans un système de base de données relationnelles.

Interface workflow	
Fonction	Assure la communication entre le système et un mécanisme de workflow existant.
Description	Cette classe d'objet est utilisée dans les fonctionnalités suivantes : • Le contrôle de légitimité Acteurs : autorité d'approbation • Le contrôle de faisabilité Acteurs : gestionnaires des ressources • La distribution des interventions Acteurs : gestionnaires des ressources • Les tests de fonctionnement d'un service Acteurs : gestionnaires des ressources, bénéficiaire • La validation de suppression d'accès Acteurs : bénéficiaire, demandeur,gestionnaire des accès Les workflows fonctionnent sous le principe d'un scénario, ce dernier comprend la liste des acteurs intervenant dans celui-ci ainsi que le ou les formulaires à compléter par ces acteurs.
Type	Spécifique car celle classe doit être adaptée au mécanisme de workflow existant dans l'organisation. Il en existe plusieurs disponible actuellement (Lotus-Notes, ...)
Interface	InitWF(Scénario : Object)→ Référence :Integer ▪ Initie un nouveau workflow
	KillWF(Référence : Integer)→ Statut :Integer ▪ Stoppe un WF en fonctionnement
	GetStatus(Référence : Integer)→ Statut :Integer ▪ Donne le statut d'un WF en fonctionnement
Attributs	Liste des workflows [0-n] : Object ▪ Liste des workflows en fonctionnement
	Liste des scénarios [0-n] : Object ▪ Liste des scénarios existants
Persistance	Les informations qualifiant les workflows seront sauvegardées dans un système de base de données relationnelles.

Interface Scheduler	
Fonction	Assure la communication entre SAGA et le système d'ordonnancement de tâches du système d'exploitation.
Description	Cette classe d'objet permet de demande l'exécution d'une action à un moment déterminé. Elle est utilisée pour assurer les fonctionnalités suivantes : ▪ Keep alive Un accès a été configuré pour la vérification régulière de l'utilisation de celui-ci. Le scheduler initie cette procédure de vérification. ▪ Echéance d'un accès Un accès a été configuré pour une durée déterminée. A l'échéance, le scheduler initie une procédure de suppression avec avertissement préalable du bénéficiaire. ▪ Activation différée Un accès a été configuré pour être activé à une date déterminée ▪ Time-out pour Workflow (si WF interne) Un WF a été déclenché. Le déroulement de celui-ci peut nécessiter l'utilisation d'un time-out (ex : un acteur ne répond pas dans les délais impartis). Le scheduler prévient le WF de cet événement.
Type	Spécifique car doit être adapté à la plate-forme sur laquelle SAGA est installé
Interface	SetEvent(Méthode CallBack : Object, Date d'exécution : Date) → Référence Event : Integer ▪ Crée un nouvel évènement
	DeleteEvent(Référence Event : Integer) → Statut : Integer ▪ Supprime un évènement
Attributs	Liste des events [0-n] : String ▪ Liste des évènements en attente d'exécution
Persistance	Les informations qualifiant les événements du scheduler seront sauvegardées dans un système de base de données relationnelles.

Configuration système	
Fonction	Assure la gestion de la configuration, c'est à dire la sauvegarde et la recherche des paramètres nécessaires aux différents modules de SAGA.
Type	Générique
Interface	Load() → Statut : Integer ▪ Charge en mémoire l'ensemble des paramètres de configuration
	Save() → Statut : Integer ▪ Sauvegarde l'ensemble des paramètres
Attributs	Paramètres [0-n] : String ▪ Liste des paramètres du système sous la forme 'nom=valeur'
Persistance	Les paramètres sont sauvegardés dans un fichier sous format XML.

Consultation	
Fonction	Permet aux utilisateurs du système la consultation des informations gérées par SAGA.
Description	Cette classe d'objet offre la possibilité aux utilisateurs de consulter des informations gérées par SAGA en temps réel. Cette recherche d'informations passe par des requêtes définies lors de la mise en place du système.
Type	Générique
Interface	RequestInfo(Request : Integer) → Response : Objet ▪ Demande l'exécution d'une requête
Attributs	Liste des requêtes [0-n] : Objet ▪ Liste des requêtes disponibles
Persistance	Les informations qualifiant les requêtes seront sauvegardées dans un système de base de données relationnelles.

Information de configuration	
Fonction	Qualifie une information de configuration d'une ressource
Type	Générique
Interface	New() → Statut :Integer ▪ Crée une nouvelle information
	Load() → Statut :Integer ▪ Charge une information existante
	Save() → Statut : Integer ▪ Sauvegarde l'information
	Delete() → Statut : Integer ▪ Suppression de l'information
Attributs	Référence : Integer ▪ Référence (identifiant) de l'information
	Nom : String ▪ Nom de l'information
	Libellé : String ▪ Libellé apparaissant dans les formulaires de demande d'accès
	Description : String ▪ Description de l'information
	Catégorie : String ▪ Catégorie de l'information
	Type : String ▪ Type de l'information (numérique…)
	Valeur par défaut : String ▪ Valeur par défaut de l'information
	Valeur minimum : String ▪ Valeur minimum de l'information
	Valeur maximum : String ▪ Valeur maximum de l'information
Persistance	Les informations qualifiant les informations de configuration seront sauvegardées dans un système de base de données relationnelles.

Sécurité	
Fonction	Assure le contrôle d'accès des différentes fonctions de SAGA.
Description	Cette classe d'objet assure les fonctionnalités suivantes : • L'authentification des utilisateurs du système • Le contrôle d'accès aux différentes fonctionnalités du système Chaque utilisateur du système se voit attribuer un profil d'utilisation. L'accès aux fonctions est contrôlée par rapport à ce profil.
Type	Générique
	Authentification(Utilisateur : Personne) → Statut :Boolean ▪ Authentification de l'utilisateur
	ControlAccess(Profil : Profil, Fonction : Methode) → Statut :Boolean ▪ Contrôle si l'accès à la fonction est autorisé pour le profil donné
Attributs	Liste des accès [Fonction : Objet] : Liste de profil[0-n] : Profil ▪ Tableau contenant les profils autorisés pour chaque fonction
Persistance	Les informations qualifiant la liste des accès autorisés pour un profil donné seront sauvegardées dans un annuaire.

Persistance	
Fonction	Assure les fonctions d'enregistrement et de lecture des informations gérées par SAGA.
Description	Cette classe d'objet a pour fonction de déterminer l'espace de stockage des informations et ensuite de procéder à l'enregistrement ou à la lecture de celles-ci. Cette classe est aussi chargée de transformer l'information 'objet' sous la forme acceptable par l'espace de stockage.
Type	Générique
Interface	Read(Adresse : Objet) → Information : Objet ▪ Renvoie l'information contenue à l'adresse donnée
	Write(Adresse : Objet, Information : Objet) → Statut : Integer ▪ Enregistre l'information à l'adresse donnée

Auditing & Alarming	
Fonction	Journalise les évènements importants et génère une alarme en cas d'événement particulier.
Description	La fonctionnalité principale de cette classe d'objet consiste à enregistrer les événements importants générés par les différents modules de SAGA dans un fichier journal. De plus, une alarme peut être générée sous certaines conditions pour signaler, par exemple, un disfonctionnement du système. Les évènements passent dans un premier filtre avant d'être enregistrés. Ils passent ensuite par un deuxième filtre qui détermine s'il y a lieu d'envoyer une alarme. Les deux filtres ainsi que la destination des logs et des alarmes sont configurés à l'aide de l'objet « Configuration du système».
Type	Générique
Interface	LoadConfiguration() → Statut : Integer ▪ Charge les informations de configuration des filtres et de la sélection de la destination des logs et des alarmes
	LogEvent() → Statut : Integer ▪ Ecrit un événement dans le journal
	SendAlarm() → Statut : Integer ▪ Envoie une alarme
Attributs	Filtre des évènements [0-n] : Integer ▪ Liste des n° d'évènements qui seront enregistrés dans le journal
	Filtre des alarmes [0-n] : Integer ▪ Liste des n° d'évènements qui génèreront une alarme
	Fichier journal : String ▪ Nom du fichier journal
	Destination alerte : String ▪ Adresse du système qui recevra les alertes
Persistance	Le format du fichier journal utilisera la norme XML.

Reporting	
Fonction	Assure les fonctions de reporting (génération de rapport) des informations gérées par SAGA.
Description	Nous suggérons d'utiliser un produit spécialisé (consulter le chapitre 'conseils d'implémentation) pour remplir les fonctions de reporting avec le système SAGA.
Type	Générique

Légitimité	
Fonction	Assure les fonctions de contrôle de légitimité d'une demande d'accès.
Description	Cette classe d'objet est chargée d'effectuer le contrôle de légitimité d'une demande d'accès. Cette classe est spécifique, ce qui signifie qu'elle doit être adaptée suivant le domaine d'application de la demande d'accès. Consulter le 8.3.6 'Contrôle de légitimité'
Type	Spécifique
Interface	New (DA : Demande d'accès) → Statut : Integer ▪ Crée un nouvelle instance du contrôle de légitimité
	SearchApprobationAuthority() → Peoples[0-n] : Liste de personnes ▪ Détermine l'autorité d'approbation chargée d'approuver la demande
	CtrlLegitimity() → Statut : Integer ▪ Contrôle si la DA est légitime.
Attributs	Méthode de contrôle : Integer ▪ Méthode utilisée pour le contrôle
	Autorité [0-n] : Liste de personnes ▪ Liste des personnes chargées de contrôler la légitimité
Persistance	Les informations nécessaires pour le contrôle de légitimé seront sauvegardées sous la forme de fichiers plats de configuration.

Faisabilité	
Fonction	Assure les fonctions de contrôle de faisabilité d'une demande d'accès.
Description	Cette classe d'objet est chargée d'effectuer le contrôle de faisabilité d'une demande d'accès. Cette classe est spécifique, ce qui signifie qu'elle doit être adaptée suivant le domaine d'application de la demande d'accès.
Type	Spécifique
Interface	New (DA : Demande d'accès) → Statut : Integer ▪ Crée un nouvelle instance du contrôle de faisabilité
	SearchGestionnaires() → Gestionnaires[0-n] : Liste de personnes ▪ Détermine les personnes chargées d'évaluer la faisabilité
	CtrlFaisability() → Statut : Integer ▪ Contrôle si la DA est réalisable techniquement
Attributs	Gestionnaires [0-n] : Liste de personnes ▪ Liste des personnes chargées d'évaluer techniquement la demande d'accès pour un service particulier
Persistance	Les informations nécessaires pour le contrôle de faisabilité seront sauvegardées sous la forme de fichiers plats de configuration.

9.3.5 Hiérarchie des classes d'objets

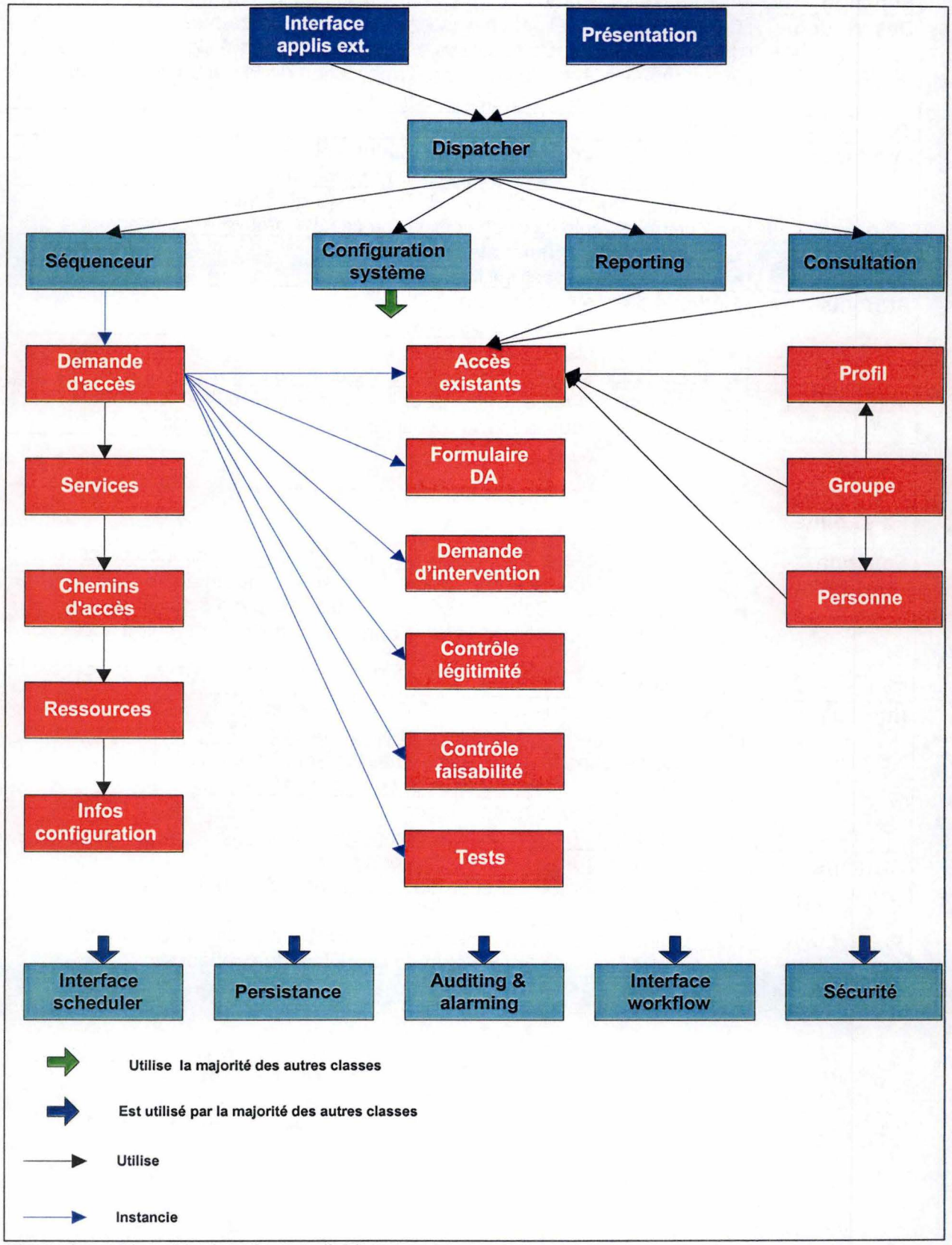

Figure 22 - Hiérarchie des classes d'objets

Notez que pour une question de clarté, seules les relations principales sont indiquées.

9.4 Conseils d'implémentation

9.4.1 Introduction

A présent que l'on a déduit les grandes lignes de l'architecture de SAGA (voir la « Figure 20 - Architecture de SAGA » – page102), Il nous reste à définir, pour chacun des couches de cette architecture, les choix techniques. Il est important de préciser que nous ne souhaitons pas fixer à tout prix ces choix, mais plutôt de donner une vue sur les différentes façons d'implémenter SAGA. Il faut noter qu'une implémentation réelle de SAGA impliquerait la prise en compte de l'environnement existant dans lequel SAGA doit s'intégrer ainsi que du ou des domaines d'application pour lesquels il faut gérer les accès. Nous tenons compte de l'aspect économique ce qui guide la plupart de nos choix vers des produits *open-source*.

9.4.2 Choix préliminaires

Nous fixons dés à présent deux points importants qui influenceront nos choix ultérieurs :

- *La liste des plate-formes sur lesquelles SAGA pourrait être installé*
  Nous souhaitons évidemment que SAGA puisse être porté sur le maximum de plate-formes existantes. Tous les choix techniques ultérieurs devront tenir compte des plate-formes cibles qui auront été choisies. Nous pensons que le portage sur les plate-formes NT et Linux constitue un minimum obligatoire.

- *Le langage de développement*
  Nous recherchons un langage objet porté sur les plate-formes définies ci-dessus et pour lequel il existe déjà un maximum d'API écrites pour les fonctions recherchées. Le langage Java semble constituer un choix judicieux.

9.4.3 Couche présentation

Cette couche assure la communication entre SAGA et le monde extérieur. Nous avons défini dans le 5.3 'Liste des acteurs' la liste des acteurs en communication avec le système SAGA. On peut tenter de classer en catégorie ces acteurs :

- *Communication interactive entre SAGA et les utilisateurs*
  Nous avons précisé dans le chapitre 6 'Liste des exigences' que l'interface utilisateur devrait être la plus légère possible. Ces deux contraintes nous ont orienté naturellement vers une solution de type « client léger » (sans *Applet* java ni d'*ActiveX*) avec navigateur web et serveur *HTTP*. Il sera possible d'utiliser un serveur HTTP existant dans l'organisation ou d'utiliser celui fourni avec le serveur d'application. Sinon nous préconisons l'utilisation d'Apache. [www.apache.org].

- *Communication indirecte entre SAGA et les utilisateurs*
  Certaines fonctionnalités de SAGA utilisent un mécanisme de workflow pour faire parvenir des formulaires électroniques à une liste prédéfinie d'utilisateurs. Nous avons signalé qu'il existait pas encore de standard normalisant la communication avec les systèmes de workflow. Il sera donc nécessaire de développer une interface spécifique. Le système de workflow existant dans l'organisation pourra être utilisé, sinon nous préconisons l'utilisation du produit *open-source* « workflow toolkit » [http://www.vivtek.com/wftk] ou « Twig » [http://twig.screwdriver.net/about.php3]

- *Communication initiée par SAGA à destination d'une application extérieure*
  Ce type de connexion est utilisée quand SAGA doit communiquer des informations avec une autre application, par exemple, s'il doit transmettre un courrier électronique ou avertir un système de monitoring d'un éventuel problème. Les protocoles utilisés dans ce cas sont divers et comprennent notamment *SMTP*, *SNMP*,... Il s'agira de créer une classe d'objet spécifique pour chaque application cliente.

- *Communication initiée par une application externe à destination de SAGA*
  Dans ce cas, SAGA répond aux requêtes émises par des applications externes, par exemple dans le cas ou une application des ressources humaines désire créer des accès pour un nouveau collaborateur. Cette communication fait appel à un mécanisme de *RPC* ou SAGA joue le rôle de serveur et les applications externes le rôle de clients. Plusieurs solutions (*CORBA, RMI*) s'offrent à nous, et en fin de compte, notre choix s'est posé sur *SOAP* (Single Object Access Protocol) car CORBA se révélait trop complexe à mettre en œuvre et RMI posait certaines contraintes au niveau de la sécurité.
  SOAP est un mécanisme de RPC qui utilise le codage d'informations XML sur un couche de transport HTTP. Il existe des clients et serveurs SOAP écrits sous différents langages (*Java, Perl, VB*,...)

9.4.4 Couche logique applicative

Nous conseillons l'utilisation d'un serveur applicatif pour la gestion de la couche logique applicative.

Un serveur applicatif est constitué de composants techniques et de composants métiers. Les composants techniques apportent un grand nombre de fonctionnalités telles que la gestion des transactions, la communication avec la couche persistance où encore, la gestion de la sécurité. Le développeur est ainsi libéré de tous les aspects techniques et peut se concentrer sur l'écriture des composants métiers.
Les composants métiers assurent les fonctions de base de l'application. Une fois déployés au sein d'un serveur d'application, ces composants vont pouvoir supporter des appels concurrents et une montée en charge. Le serveur d'application dispose de consoles qui permettent une administration centralisée. La fiabilité et la disponibilité sont prise en compte par les possibilités étendues de tolérance de panne et d'architecture redondante qu'offre ces serveurs applicatifs.

On pourra utiliser le serveur applicatif déjà présent dans l'organisation. Sinon, nous préconisons l'usage de serveurs applicatifs open-source. On peut citer les produits Enhydra [www.enhydra.org] et Zope [www.zope.org]

Les personnes réfractaires à l'utilisation d'un serveur applicatif pourront utiliser une architecture plus classique basée sur les *servlets*.

9.4.5 Couche persistance

Nous avons indiqué dans le chapitre 9.2 'Architecture', que la couche persistance était en fait constituée de trois espaces de stockage différents :

- Un annuaire
  Notre préférence va vers l'utilisation d'un annuaire LDAP. L'accès à cet annuaire est normalisé et il existe de nombreux outils pour la mise à jour de celui-ci. On pourrait ainsi externaliser la gestion des personnes, des groupes et des profils, par exemple, déléguer au service des ressources humaines cette gestion.

  Il est à noter que la délégation de la gestion des personnes, groupes et profil à un annuaire LDAP rend SAGA aveugle des modifications effectuées dans cet annuaire. Ce qui pose des problèmes de cohérence. Nous avons indiqué dans le 8.3.5 'Gestion des événements et de leurs impacts sur les accès' que l'ajout d'une personne dans un groupe pouvait générer une modification des accès. Si cet ajout est effectué directement dans l'annuaire LDAP, SAGA n'est pas au courant de cette modification et les accès existants de cette personne sont incohérents.
  Il existe plusieurs solutions à ce type de problème :
  - effectuer les modifications de l'annuaire au travers d'un proxy spécifique qui signalera à SAGA les évènements importants pour lui.
  - installer un *plug-in* dans l'annuaire pour effectuer la même fonction que ci-dessus
  - effectuer des contrôles réguliers de l'annuaire LDAP pour y détecter les changements et avertir SAGA.

  La solution à adopter dépendra de l'environnement existant de l'organisation ainsi que des contraintes de performances et des délais acceptables pour la mise à jour des accès.

  S'il n'existe pas encore de système d'annuaire dans l'organisation, nous préconisons d'utiliser OpenLDAP [http://www.openldap.org/], un produit open-source.

- Un système de base de données relationnelles
  Il existe de nombreux systèmes de base de données, il en existe également probablement déjà dans l'organisation où il faudrait implémenter SAGA. Les contraintes principales pour le choix du SGBD sont de :
  - supporter les procédures stockées et les triggers (pour la mises à jour automatique des accès)
  - supporter l'aspect transactionnel (cohérence des informations stockées)
  - posséder un connecteur JDBC permettant à la logique applicative écrite en JAVA d'accéder à la base de données

  Dans le domaine de l'open-source, on peut citer le produit PostgreSQL [http://www.postgresql.org/] qui répond à toutes ces contraintes.

- *Des fichiers plats sous forme XML*
  Le principal avantage des fichiers plats est la simplicité d'édition et de consultation. Le format XML fournit un format adapté aux formulaires. De plus, il existe de nombreux « parsers » dans de nombreux langages.

9.4.6 Reporting

La notion de reporting signifie pour nous la génération de divers rapports sur les informations traitées et stockées par SAGA. Nous avons choisi, toujours avec l'idée de ne pas réinventer la roue, d'utiliser si possible, des produits existants. Nous avons vu qu'il existait en fait trois espaces de stockage pour ces informations.

- *Annuaire LDAP*
  Les applications permettant d'interroger des annuaires LDAP sont peu nombreuses, nous préconisons le produit commercial Calandra [http://www.calandra.com/]

- *SGBD*
  S'il n'est pas possible d'utiliser une application d'interrogation existante dans l'organisation, nous préconisons un produit commercial Crystal-Report [http://www.seagate.com/]

- *Fichiers plats au format XML*
  Il existe peu d'applications qui permettent d'effectuer des interrogations sur des fichiers contenant des informations sous format XML. On consultera à ce sujet le document de travail publié par le WWW Consortium sur les possibilités d'interrogation des informations au format XML [http://www.w3.org/XML/Query]. Ce document aboutira certainement à des applications très prochainement.

Cette approche a comme inconvénient qu'il est nécessaire d'utiliser trois outils différents pour générer les rapports car un outil ne peut couvrir seul simultanément les informations de ces espaces de stockage.

9.4.7 Sécurité

La sécurité du système SAGA repose sur l'authentification des utilisateurs, le contrôle des actions que ces derniers entreprennent et le contrôle des informations qui leur sont présentées. Nous voyons trois possibilités pour réaliser ces contrôles :

- Implémenter dans SAGA les fonctionnalités pour réaliser ces contrôles
- Utiliser un serveur applicatif pour développer l'application SAGA et qui dispose déjà des fonctionnalités nécessaires pour ces contrôles. La discussion sur les serveurs applicatifs figure dans le chapitre 9.4.4 'Couche logique applicative ' et précise en détail le fonctionnement de ces serveurs.
- Déléguer à une application externe la sécurité de SAGA.
  En effet, SAGA étant une application de type Web, nous pouvons déléguer cette gestion à une application qui agit en tant que '*proxy http*'. Ce qui signifie, qu'elle va agir en tant qu'intermédiaire entre les utilisateurs et le serveur HTTP du système SAGA. Cette application pourra contrôler tout le trafic entrant et sortant entre l'utilisateur et l'application SAGA. Cette application va chercher dans un annuaire *LDAP* les informations concernant les utilisateurs et peu ainsi assurer l'authentification. Chaque fonctionnalité de SAGA à sécuriser correspond à un URL particulier. Le proxy, en pratiquant un filtrage sur ces URL's peut ainsi contrôler l'accès à ces fonctionnalités.

  Nous n'avons pas trouve de produit open-source pour cette application, nous citerons un produit commercial SiteMinder [www.netegrety.com] dont voici le schéma de principe :

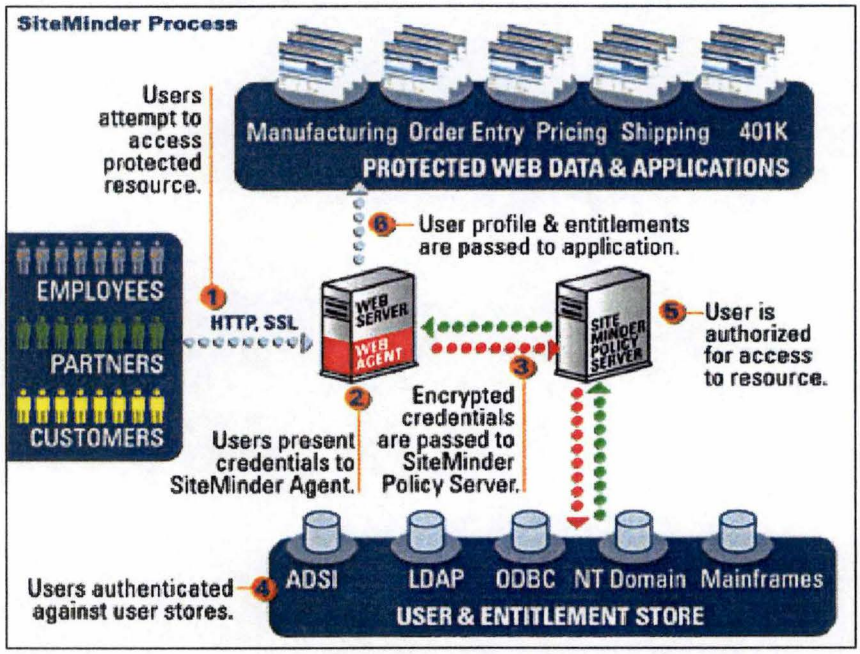

Figure 23 - Principe de fonctionnement de SiteMinder (Source : NetIntegrity)

Nous ne souhaitons pas gérer dans SAGA la sécurité du système. Si l'organisation dispose d'une application de type SiteMinder, nous préconisons d'utiliser cette application, sinon l'utilisation d'un *serveur applicatif* permettrait de rencontrer les exigences formulées pour la sécurité du système.

9.4.8 Architecture détaillée de SAGA

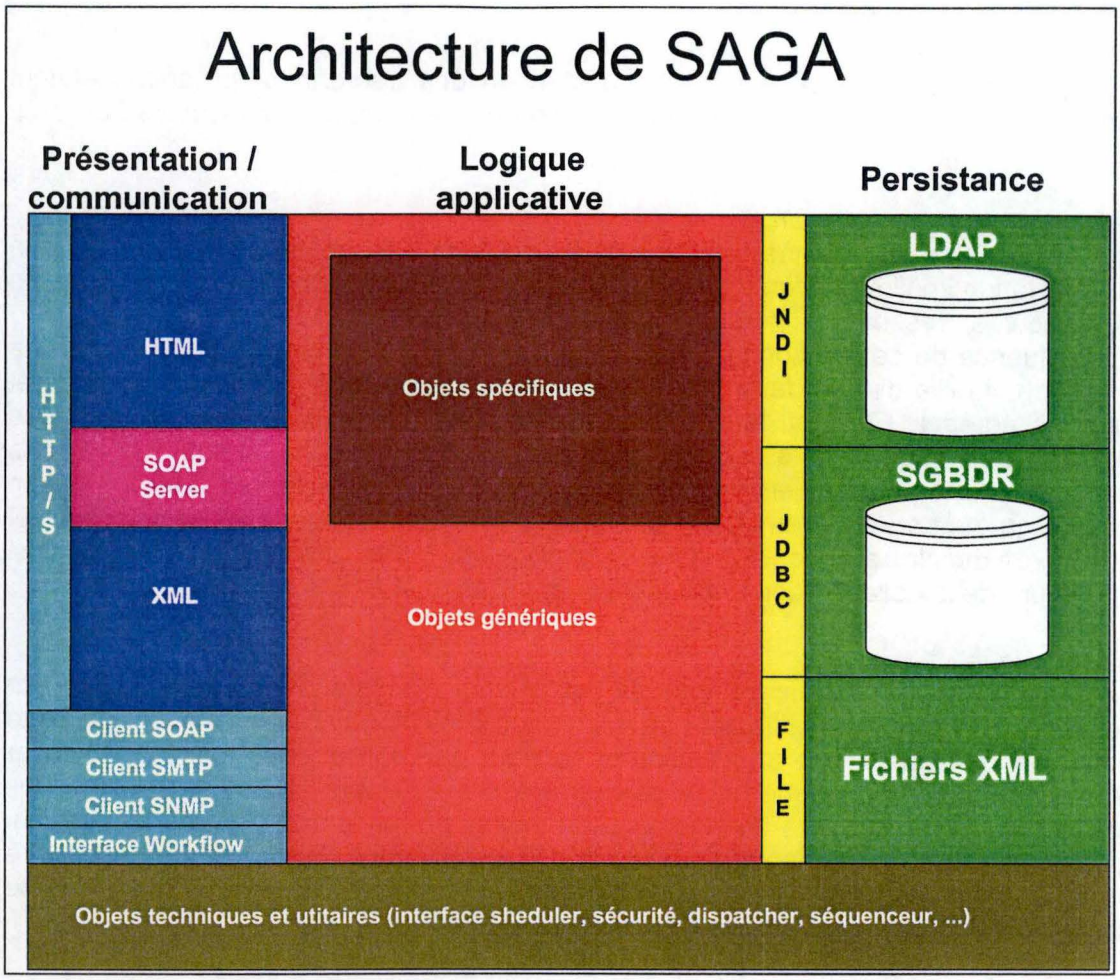

Figure 24 - Architecture détaillée de SAGA

9.5 *Conseils de mise en œuvre*

L'unique objectif de ce chapitre est d'attirer l'attention des personnes qui souhaiteraient mettre en œuvre SAGA au sein de leur organisation sur les aspects particuliers liés à ce système.

1) *gestion centralisée et inventaire*
   Le fait que SAGA se positionne comme le système unique de gestion des accès au sein de l'organisation implique qu'on y retrouve la liste complète et à jour des objets impliqués dans la sécurité : ressources, personnes, gestionnaires, service, ...
   La conséquence de ce positionnement est que, pour mettre en œuvre SAGA dans une organisation, il faille d'abord faire un recensement précis de tous ces objets existants et ensuite les agréger. En effet, le fait qu'un même objet (par exemple les coordonnées d'une personne) puisse être partagé par plusieurs applications, induit parfois des redondances. Il se peut également que certains objets soient gérés par une application spécifique qui ne sera pas remplacée par SAGA. Il est donc important, en réalisant cet inventaire, de définir clairement les rôles de chaque application par rapport à SAGA ainsi que la nature de la collaboration entre chaque application et SAGA.

2) *passage unique*
   Le système SAGA n'est efficace que s'il est le seul système dans l'organisation par lequel les accès sont gérés. Il existe probablement déjà des systèmes au sein de cette organisation qui permettent aux différents acteurs de réaliser leur mission. Imposer SAGA comme système unique ne va pas être simple au début ; il risque d'y avoir des réticences liées au fait que le changement fait toujours peur et que certains acteurs ont peut-être des habitudes qui sont bien encrées. Nous pensons donc qu'il est nécessaire qu'il y ait une volonté politique forte d'imposer SAGA comme système unique au sein de l'organisation.

3) *déploiement progressif ou déploiement en une fois*
   Comme pour tout projet, le déploiement de SAGA peut se faire soit en une seule fois, soit progressivement. Le déploiement en une seule fois a comme principal inconvénient qu'il faut avoir un inventaire complet de tous les objets existants, avoir convaincu et formé tous les acteurs avant de pouvoir mettre le système en activité. Le déploiement progressif semble par contre être une alternative intéressante il faut cependant faire attention à la manière dont on découpe les phases de reprise de l'existant. Nous conseillons d'avoir une approche transversale plutôt que verticale ; c'est à dire reprendre service par service(au sens SAGA du terme) plutôt que de reprendre le tout en limitant les fonctionnalités disponibles (par exemple : ne pas faire de contrôles sur la demande dans un premier temps). Cette approche a le mérite de concerner moins de personnes en une fois et donc de permettre une acclimatation progressive des acteurs au système. Elle a aussi un effet de bord positif (on peut l'espérer) qui est de promouvoir la publicité pour le système par le bouche-à-oreille. Nous pensons qu'il est plus intéressant de satisfaire complètement un petit nombre de personnes que de satisfaire à moitié l'ensemble complet des personnes.

4) *distribution des rôles*
   Un autre aspect important dont il faut tenir compte lors de la mise en œuvre de SAGA est la distribution des rôles aux différents acteurs. Pour que le système porte pleinement ses fruits, il faut impérativement que chacun sache quel rôle il y joue et qu'on lui explique quels sont les autres rôles et les différentes interactions entre ceux-ci. Il est important qu'une relation de confiance s'établisse entre les différents acteurs au travers du système.

10 Illustration de la généricité

10.1 Introduction

L'objectif de ce chapitre est principalement de vérifier avec des cas réels empruntés à différents domaines que les modèles génériques des données et des traitements de SAGA couvrent bien tous les aspects de la problématique de gestion des accès. Cette illustration donne également la possibilité au lecteur de bien comprendre le positionnement des concepts de SAGA dans les différents domaines abordés.

En ce qui concerne les traitements, nous allons comparer notre scénario générique d'une demande de création d'accès - décrit dans le chapitre 8.2.4. 'Scénario générique d'une demande d'accès' - avec les scénarios spécifiques que nous avons pu dresser grâce aux interviews des personnes impliquées dans les trois domaines d'applications de référence. Pour rappel, il s'agit de demandes concernant :

- Un accès à un emplacement de parking
- Un accès à des ressources informatiques
- Un accès à des locaux sécurisés

En ce qui concerne les données, nous allons vérifier que nous retrouvons chaque concept du modèle des données dans les cas présentés.

10.2 Demande d'une carte riverain (parking)

10.2.1 Contexte

Un riverain souhaite bénéficier d'un emplacement de parking dans une zone piétonne du centre ville namurois qui est protégée par des bornes amovibles. Il remplit un formulaire papier qu'il transmet au service « stationnement » de la Police Communale de Namur. Le service « stationnement » vérifie si le nombre d'emplacements déjà réservés dans cette zone n'atteint pas le quota maximum déterminé et si le bénéficiaire est bien domicilié dans la rue correspondant à l'emplacement demandé. D'autre part, il vérifie que le type d'emplacement permet le stationnement de la catégorie du véhicule du bénéficiaire et que le riverain ne dispose pas d'un garage situé dans un rayon d'un kilomètre par rapport à son domicile. Le service « stationnement » prépare alors la carte magnétique du demandeur et communique au service de la régie urbaine de l'équipement le code de cette carte ainsi que la référence de la zone piétonne. La régie urbaine de l'équipement configure alors chaque borne d'entrée de la zone piétonne en ajoutant le code de la carte à la liste déjà présente. La régie urbaine signale ensuite la fin de l'intervention au service « stationnement » de la Police Communale qui invite alors le riverain à venir chercher sa carte magnétique. Le service « stationnement » archive le formulaire de demande pour pouvoir effectuer le contrôle au niveau des cartes octroyées et en transmet une copie à la comptabilité.
Avec cette carte, le riverain a le droit d'occuper tout emplacement de parking situé dans la zone en question.

10.2.2 Données

- *Service* : les emplacements de parking dans le centre ville Namurois constituent le service. Remarquons qu'il y a plusieurs zones piétonnes distinctes dans le centre ville Namurois ; chacune devrait constituer un service différent si on veut distinguer l'accès à une zone de l'accès à une autre.

- *Profil* : si l'octroi de cet accès était systématique (et gratuit !) pour chaque riverain de la zone piétonne, on pourrait créer un profil « riverain » qui donnerait automatiquement cet accès à tout riverain.

- *Groupe* : une autre alternative pour donner accès à une zone piétonnière à tous les riverains serait de créer un groupe « riverains de la zone Z » et de donner accès à ce groupe au service correspondant aux emplacements de cette zone Z.

- *Accès* : on peut considérer que la copie du formulaire gardé pour effectuer le contrôle au niveau des cartes octroyées est une trace de l'accès accordé.

- *Demande d'accès* : le formulaire de demande de carte matérialise la demande d'accès

- *Demandeur* : il s'agit du riverain mais cela pourrait être son représentant légal

- *Bénéficiaire* : le riverain. Notons que le demandeur et le bénéficiaire sont une seule et même personne dans ce cas-ci

- *Groupe fonctionnel* : dans notre exemple, il serait judicieux de considérer le « service stationnement » et la « régie urbaine » comme deux groupes fonctionnels.

- *Ressource* : dans cet exemple, on peut considérer qu'il y a 2 ressources :
  - l'appareil qui permet de fabriquer la carte d'accès
  - les bornes amovibles : de la même manière, l'ensemble des bornes amovibles délimitant la zone constitue une seule et même ressource. Il est cependant loisible au gestionnaire d'accès de définir une ressource pour chaque borne ; ce qui influence évidemment la composition des chemins d'accès

- *Chemin d'accès* : dans cet exemple, il y a autant de chemins d'accès qu'il n'y a de zones piétonnières protégées par des bornes (si on veut individualiser l'accès à ces zones). Dans ce cas, on associe un seul chemin d'accès à chaque service.
  Le chemin d'accès est composé de la machine de génération des cartes et de l'ensemble des bornes amovibles. Notons que dans ce cas particulier, la notion de chemin d'accès n'illustre pas un cheminement logique au travers des différentes ressources mais est bien constituée de la liste de toutes les ressources impactées pour donner accès au service. On pourrait dire que la machine de génération des cartes est considérée comme le point d'accès, car c'est la première ressource sur laquelle il faut intervenir ; la ressource cible étant constituée de l'ensemble des bornes amovibles de la zone.
  Remarquons que si le gestionnaire d'accès choisissait de considérer chaque borne comme une ressource séparée, alors il y aurait autant de chemins d'accès que de bornes et on associerait à un même service tous les chemins d'accès impliquant les bornes d'une même zone.

- *Information de configuration* : outre les informations fournies lors de la demande sur l'identité du bénéficiaire et le service demandé, chaque ressource pourrait nécessiter pour sa configuration des informations spécifiques. Ce n'est pas le cas dans cet exemple-ci. Remarquons que des informations de configuration sont nécessaires au niveau du service : dans notre exemple, la catégorie du véhicule.

- *Opération sur l'accès* : dans notre exemple, il s'agit d'une création d'accès

- *Demande d'intervention* : il y a une demande d'intervention pour chaque ressource impactée par l'accès au service. Dans cet exemple-ci une demande d'intervention pour la configuration des bornes (ressource= bornes amovibles) est envoyée à la régie urbaine et une demande d'intervention pour la fabrication de la carte est envoyée au service « stationnement » (ressource= emplacements de parking).

- *Autorité d'approbation* : on peut considérer que l'autorité d'approbation est constituée du groupe fonctionnel « service stationnement ».

- *Niveau d'accès et action* : même si l'exemple n'en fait pas mention, on peut très bien imaginer que certaines personnes de l'administration communale aient accès au système SAGA pour y consulter les demandes en cours de traitement mais qu'elles n'aient pas le pouvoir de les clôturer. Il pourrait donc y avoir plusieurs niveaux d'accès aux informations de ce domaine. Un niveau d'accès « gestionnaire » pour les personnes des services intervenants qui permet d'entreprendre des actions de consultation et de modification. Et un autre niveau d'accès (par exemple « administration ») qui ne permette que l'action de consultation.

- *Gestionnaire de ressource* :
  - le service « stationnement » est gestionnaire de la ressource « emplacements de parking »
  - la régie urbaine est gestionnaire de la ressource « bornes amovible »

- *Gestionnaire d'accès* : le « service stationnement » semble se positionner comme gestionnaire d'accès pour le « service emplacements de parking dans le centre ville Namurois ».

10.2.3 Scénario

- *Saisie de la demande*
  - *Analyse d'impact*
    Le seul chemin d'accès associé au service nous donne la liste des ressources impactées : la machine à générer la carte et les bornes amovibles.

  - *Génération du formulaire de demande d'accès (FDA)*
    Le FDA est commun à toutes les demandes pour ce service car les types d'informations sont fixés lors de la mise en place du service.

- *Contrôles*
  - *Contrôle de validité des informations saisies*
    SAGA vérifie, à la place de l'agent du service « stationnement » de la Police Communale de Namur, que le formulaire est complet et cohérent.

  - *Contrôle de légitimité*
    Le contrôle de légitimité consiste à vérifier l'identité du demandeur, vérifier le type de véhicule et vérifier si le demandeur dispose d'un garage dans un rayon d'un kilomètre. Dans ce cas, l'autorité d'approbation est constituée de l'agent du service « stationnement » de la Police Communale de Namur.

  - *Contrôle de faisabilité*
    Ce contrôle consiste à vérifier le nombre de places disponibles dans la zone en question, pour cela, SAGA consulte la liste des accès existants pour ce service.

- *Intervention*
  - *Génération des demandes d'intervention*
    Deux interventions sont nécessaires : la création de la carte magnétique et la configuration des bornes amovibles. Il y aura donc deux demandes d'intervention (DI).

  - *Distribution des demandes d'intervention*
    La demande de création de la carte magnétique est transmise au service « stationnement » de la Police Communale de Namur. La demande de configuration des bornes amovibles est transmise à la régie urbaine de l'équipement.

- *Clôture de la demande*
  - *Mise à jour des accès existants*
    Ce nouvel accès vient compléter la liste des accès existants à cette zone.

  - *Vérification du bon fonctionnement de l'accès*
    Pas d'application dans ce cas

  - *Retour d'information*
    Cette action consiste à envoyer un courrier au riverain signalant qu'il peut venir recevoir sa carte magnétique.

10.3 Demande d'accès à des ressources informatiques (domaine IT)

10.3.1 Contexte

Un employé souhaite bénéficier à partir de son poste de travail d'un accès Internet. Le responsable de la sécurité reçoit sa demande écrite. Il détermine (il les connaît par habitude) les composants impactés par cette demande. Il s'agit du navigateur web du poste de travail du bénéficiaire et du firewall. Il consulte ensuite le responsable hiérarchique du bénéficiaire qui lui confirme que cette demande est valide. Le responsable de la sécurité envoie alors les informations nécessaires à l'équipe déploiement afin qu'elle configure la navigateur web du bénéficiaire et également à l'équipe système chargée de configurer le firewall. Ces équipes effectuent les interventions nécessaires et avertissent le responsable de la sécurité quand celles-ci sont terminées. Le responsable de la sécurité avertit enfin le bénéficiaire de l'accès demandé que cet accès est disponible et lui communique le code d'accès. Le responsable de la sécurité garde un historique de tous les accès qui sont octroyés, ainsi que des raisons qui les ont justifiés.

10.3.2 Données

- *Service* : Internet est le service auquel on peut demander accès
- *Profil* : pas d'application
- *Groupe* : pas d'application
- *Accès* : la notion d'historique des accès octroyés avec pour chacun d'eux ses justificatifs
- *Demande d'accès* : la demande écrite reçue par le responsable de la sécurité peut matérialiser la demande d'accès
- *Demandeur* : il s'agit de l'employé
- *Bénéficiaire* : il s'agit de l'employé. Ici aussi le demandeur et le bénéficiaire sont la même personne.
- *Groupe fonctionnel* : pas d'application
- *Ressource* : dans cet exemple, on peut considérer qu'il y a 2 ressources :
  - le poste de travail du bénéficiaire
  - le firewall
- *Chemin d'accès* : le chemin d'accès est constitué du poste de travail du bénéficiaire(point d'accès) et du firewall(ressource-cible)
- *Information de configuration* : la ressource firewall exige que l'on fournisse l'adresse IP du poste de travail du bénéficiaire
- *Opération sur l'accès* : dans notre exemple, il s'agit d'une création d'accès
- *Demande d'intervention* : une demande d'intervention sur le poste de travail du bénéficiaire est envoyée à l'équipe déploiement et une demande d'intervention sur le firewall est envoyée à l'équipe système

- *Autorité d'approbation* : le responsable hiérarchique du bénéficiaire
- *Niveau d'accès et action* : même considération que dans l'exemple précédent
- *Gestionnaire de ressource* :
  - l'équipe déploiement est gestionnaire de la ressource « poste de travail »
  - l'équipe système est gestionnaire de la ressource « firewall »
- *Gestionnaire d'accès* : le rôle du gestionnaire d'accès est joué par le responsable de la sécurité

10.3.3 Scénario

- *Saisie de la demande*
  - *Analyse d'impact*
    Le poste de travail du bénéficiaire est considérée comme le point d'accès. La ressource cible (le firewall) est commune à tous les chemins d'accès pour ce service. Le choix du chemin d'accès donc de l'identité du bénéficiaire.
  - *Génération du formulaire de demande d'accès (FDA)*
    Le FDA est commun à toutes les demandes pour ce service car les types d'informations sont fixés lors de la mise en place du service.
- *Contrôles*
  - *Contrôle de validité des informations saisies*
    Vérification si le formulaire est complet et son contenu cohérent.
  - *Contrôle de légitimité*
    Le contrôle de légitimité consiste à demander l'accord de l'autorité d'approbation. L'autorité d'approbation est composée dans ce cas du responsable hiérarchique du bénéficiaire. Ce dernier complète et renvoie le formulaire transmis par un mécanisme de work-flow.
  - *Contrôle de faisabilité*
    Pas d'application.
- *Intervention*
  - *Génération des demandes d'intervention*
    Deux demandes d'intervention seront générées : la première concerne la configuration du navigateur web du bénéficiaire et la seconde concerne la configuration du firewall.
  - *Distribution des demandes d'intervention*
    La première demande d'intervention est transmise à l'équipe déploiement et la seconde à l'équipe système.

- *Clôture de la demande*
  - *Mise à jour des accès existants*
    Cette action correspond à l'actualisation des informations détaillant les accès existants à ce service.

  - *Vérification du bon fonctionnement de l'accès*
    Pas d'application

  - *Retour d'information*
    La disponibilité de l'accès ainsi que le code d'accès sont transmis par courrier électronique ou papier au bénéficiaire.

10.4 Demande d'accès à des locaux sécurisés

10.4.1 Contexte

Un employé souhaite accéder à la salle machine d'une organisation bancaire. Celle-ci est protégée par un sas d'entrée du couloir qui mène à la salle - dont l'ouverture est actionnée par une carte magnétique combinée à un code d'accès – et ensuite par une porte blindée activée par un code d'accès personnel. Ces deux éléments de protection physique sont gérés par une application informatique de gestion des accès physiques qui est livrée par le fournisseur de matériel.
Une demande d'accès est transmise au gestionnaire de l'accès aux locaux via courrier électronique par le responsable hiérarchique de la personne bénéficiaire. Le gestionnaire d'accès vérifie l'identité du demandeur et contrôle si l'accès requis correspond à la fonction du bénéficiaire. Il ajoute ensuite le nom de cette personne dans le groupe ayant le droit d'accéder à la salle machine. Cette définition est faite dans l'application de gestion des accès physiques. Il prévient enfin par courrier électronique le bénéficiaire et son responsable hiérarchique de la disponibilité de cet accès. Il communique par la même voie le code d'accès au bénéficiaire.

10.4.2 Données

- *Service* : il s'agit de demander accès à la salle machine

- *Profil* : pas d'application mais on pourrait imaginer que tous les gestionnaires systèmes aient systématiquement accès à la salle machine ; dans ce cas on créerait un profil « gestionnaire système » qui donnerait accès au service « salle machine »

- *Groupe* : pas d'application mais rien n'empêche le gestionnaire d'accès de rassembler dans un groupe toutes les personnes qui ont accès à la salle machine

- *Accès* : l'application de gestion des contrôles d'accès physique archive les accès octroyés aux différentes personnes ; SAGA pourrait le faire à sa place ou en complément

- *Demande d'accès* : au lieu d'envoyer un formulaire de demande par courrier électronique, le demandeur pourrait utiliser l'interface de SAGA pour faire sa demande

- *Demandeur* : il s'agit du responsable hiérarchique de l'employé

- *Bénéficiaire* : l'employé

- *Groupe fonctionnel* : pas d'application

- *Ressource* : dans cet exemple, on peut considérer qu'il y a 2 ressources :
  - le sas d'entrée
  - la porte à code

- *Chemin d'accès* : le chemin d 'accès est constitué du sas d'entrée (point d'accès) et de la porte blindée de la salle machine (ressource-cible)

- *Information de configuration* : aucune autre information que celle qualifiant le demandeur, le bénéficiaire ou le service n'est requise

- *Opération sur l'accès* : dans notre exemple, il s'agit d'une création d'accès

- *Demande d'intervention* : deux demandes d'intervention partent vers le gestionnaire de l'application de configuration des accès physiques

- *Autorité d'approbation* : il s'agit du gestionnaire d'accès

- *Niveau d'accès et action* : même considération que dans l'exemple précédent

- *Gestionnaire de ressource* :
  Dans cet exemple, les deux ressources sont gérées par un même gestionnaire : le gestionnaire du système de gestion des accès physiques qui est aussi le gestionnaire d'accès

- *Gestionnaire d'accès* : il s'agit du gestionnaire du système de gestion des accès physiques

10.4.3 Scénario

- *Saisie de la demande*
  - *Analyse d'impact*
    Le seul chemin d'accès associé au service nous donne la liste des ressources impactées : le sas d'entrée et la porte blindée de la salle machine.

  - *Génération du formulaire de demande d'accès (FDA)*
    Le FDA est commun à toutes les demandes pour ce service car les types d'informations sont fixés lors de la mise en place du service.

- *Contrôles*
  - *Contrôle de validité des informations saisies*
    SAGA vérifie si le formulaire est complet et son contenu cohérent.

  - *Contrôle de légitimité*
    Le contrôle de légitimité consiste à vérifier si la fonction du bénéficiaire dans l'organisation lui permet d'avoir accès à ce local et à vérifier si le demandeur est bien le supérieur hiérarchique du bénéficiaire.

  - *Contrôle de faisabilité*
    Pas d'application.

- *Intervention*
  - *Génération des demandes d'intervention*
    Une demande d'intervention est générée pour chaque ressources : le sas d'entrée et la porte blindée de la salle machine

  - *Distribution des demandes d'intervention*
    La demande d'intervention est transmise au gestionnaire de l'application de contrôle d'accès des locaux sécurisés qui est le gestionnaire des deux ressources impactées.

- *Clôture de la demande*
  - *Mise à jour des accès existants*
    Cette action correspond à l'actualisation des informations détaillant les accès existants à ce service.

  - *Vérification du bon fonctionnement de l'accès*
    Pas d'application

  - *Retour d'information*
    Le bénéficiaire reçoit un courrier électronique lui signalant la disponibilité de son accès. Son code d'accès est également contenu dans ce courrier. Le supérieur hiérarchique du bénéficiaire est également averti de la disponibilité de l'accès demandé.

10.5 Conclusions

Nous pouvons constater que, pour les trois domaines de référence, les actions entreprises peuvent s'intégrer dans notre scénario générique. Il faudra, bien sur, devoir adapter les classes d'objets spécifiques au domaine en question. Il en va de même pour les concepts du modèle des données qui semblent couvrir tous les aspects opérationnels de la gestion des accès dans ces trois domaines. SAGA est générique dans le sens ou il couvre bien la problématique de gestion des accès appliquée aux différents domaines de références.

11 Conclusion

Dans ce travail nous avons étudié la problématique de la gestion des accès de manière globale. A cette fin, nous avons défini et appliqué une méthode qui, à partir des exigences exprimées dans différents domaines d'application, nous a permis de concevoir un système d'information qui rencontre celles-ci. Cette méthode assure la traçabilité des choix et des transformations qui ont été opérés.

Au fur et mesure de notre analyse du système, nous avons réalisé la complexité du sujet et la difficulté d'en déterminer les limites. Nous avons donc choisi de nous focaliser sur un double objectif :

- Appliquer une méthodologie qui nous permette de transformer le problème en solution implémentable. Nous avons obtenu ceci en composant différentes méthodes vues aux cours.

- Étudier la généricité de la problématique de la gestion des accès; c'est à dire étudier la manière dont nous pouvions la généraliser à plusieurs domaines et en dégager les concepts communs.

Nous restons persuadés que SAGA offre des possibilités qui ont une valeur ajoutée par rapport aux systèmes existant sur le marché. Une étude comparative approfondie devrait idéalement étayer cette supposition mais, au vu des contacts professionnels que nous avons déjà pu prendre, il semble bien qu'il y ait un réel intérêt pour cette solution dont les principaux atouts sont :

- de pouvoir s'appliquer à de multiples domaines que l'on peut retrouver simultanément au sein d'une même organisation

- d'intégrer la gestion des demandes d'accès

- de prendre en charge les aspects principaux de la gestion des accès, en ce compris les événements qui influencent les accès

- permettre la centralisation de la gestion des accès en offrant un système unique qui soit adapté à chacun des utilisateurs de SAGA

- de simplifier et fiabiliser les tâches de chacun en redistribuant la complexité dans les mains des personnes les plus aptes à la gérer

Nous avons pu constater que la recherche de la généricité induit une certaine complexité, ce qui est en contradiction avec notre objectif de simplification. Cette complexité se traduit par la nécessité de réaliser une configuration importante du système au moment du déploiement.

Dans son état actuel, la communication entre SAGA et les ressources est réalisée via les gestionnaires de ressources. A ce propos, nous pensons qu'il serait intéressant d'étudier, en guise de prolongement à ce mémoire, la possibilité de réaliser les interventions directement sur les ressources.

12 Bibliographie

12.1 Références bibliographiques

Professeur Eric Dubois, « Conception des systèmes d'information », *Partie 3 : L'approche d'analyse et de conception d'un système d'information*, 1998, pages 21 et 24.

Professeur Naji Habra, « Cours de génie logiciel », *Chapitre 5 : La Conception : technique et méthodes*, 1999, slide 17.

Gérard SACCONE , « Synthèse du rapport sur "le workflow dans les progiciels en France" », 1997, accessible via le site [http://www.gls-conseil.fr/].

Serge Miranda, Anne Ruols, « Client-serveur », Editions Eyrolles, *chapitre 5 : Une taxinomie des architectures client-serveur de données.*

Pierre-Alain Muller, « Modélisation objet avec UML » ,Editions Eyrolles, *chapitre 5 : Etude de cas.*

Andrew Patzer, « Programmation Java coté serveur », Editions Eyrolles, *chapitre 8 : Connexion aux bases de données, chapitre 22 : Utilisation de Java et LDAP.*

Stephen Asbury, « Linux en entreprise », Editions Eyrolles, *chapitre 8 : Système de sauvegarde de données au format XML.*

12.2 Références Internet

SOAP (Simple Object Access Protocol)
www.soaprpc.com
http://www.w3.org/TR/soap12/
http://www.soap-wrc.com/webservices/default.asp
http://msdn.microsoft.com/msdnmag/issues/0300/soap/soap.asp

WAPI (Workflow Application Interface)
www.aiim.org/wfmc

Middleware :
http://www.soforum.com/library/middleware.shtml
http://www.transarc.ibm.com/Library/documentation/websphere/WAS-EE/fr_FR/html/atshak/atshak09.htm
http://www.mtic.pm.gouv.fr/dossiers/documents/intranet_et_si.shtml

Serveur d'applications :
http://www.mediadev.fr/fr/download/wp_appserver.pdf
www.zope.org
www.enhydra.org

LDAP (Lightweight Directory Access Protocol) :
www.openldap.org
www.stanford.edu/~hodges/talks/EMA98-DirectoryServicesRollout/Steve_Kille/index.htm

SGBD :
www.postgresql.org

Workflow :
www.webdav.org
http://twig.screwdriver.net
http://www.vivtek.com/wftk/

XML (eXtensible Markup Language) :
http://www.w3.org/XML/Query
http://www.dsml.org/

Java :

http://castor.exolab.org/

13 Annexes

ANNEXE 1 : Formulaire d'enquête

QUESTIONNAIRE

1 Contexte

Dans le cadre de notre mémoire de licence en informatique, nous voulons réaliser une étude complète sur la problématique de gestion centralisée des accès. Le travail consiste à faire le tour de la question en abordant cette problématique de manière générale ; c'est à dire pas spécifiquement au domaine IT.
Pour compléter notre expérience en la matière, nous souhaitons recueillir l'avis de personnes comme vous qui sont, dans leur milieu professionnel, confrontées à la problématique de gestion des accès.

2 Objectifs du questionnaire

Ce questionnaire a donc pour but de rassembler un ensemble représentatif d'avis sur la question de la gestion des accès.
L'objectif est de pouvoir alimenter notre travail au niveau de l'étude de l'existant et d'en déduire les besoins des différents acteurs du système.
Il sera proposé à trois types d'acteurs :

- le gestionnaire d'accès (centralisé ou spécialisé)
- le demandeur d'accès
- le gestionnaire de composant

3 Procédure

3.1 Modus operandi

Un rendez-vous est pris avec chaque personne interrogée.
Le questionnaire est envoyé à cette personne une semaine avant l'entrevue, à des fins de préparation.
L'entrevue consiste en une séance de maximum 30 minutes pendant laquelle les questions sont passées en revues et une réponse est demandée.
Le temps de préparation pour le sujet est estimé à 30 minutes

3.2 Personnes interrogées

- Responsables de la sécurité et gestionnaires d'accès : Marie-Louise M.(système central), Jean-Luc B.(systèmes distribués et firewalling), Robert D.(gestionnaire des badges d'accès), Guy R. gestionnaire des emplacements de parking à Namur.
- Demandeurs : collègues des services de développement, représentant de la GRH, collègues de domaines non-IT
- Gestionnaires de composant : collègues des groupes WAN, LAN, UNIX, NT server, huissier(ou dispaching)
- En complément, la personne chargée (Claude) de la création de certains workflows de gestion de demandes

3.3 Comment répondre ?

Le plus efficace, si vous en avez le temps, serait d'éditer le fichier de questions et d'y insérer, après chaque question, votre réponse. Si vous ne pouvez pas le faire, vos réponses aux questions seront notées lors de l'entrevue. Dans ce cas, il conviendrait de préparer sur le formulaire papier, les grandes lignes de vos réponses.

Un tout grand merci d'avance pour le temps et l'énergie que vous voudrez bien consacrer à répondre à ce questionnaire. Votre soutien nous est précieux.

4 Questions pour les gestionnaires d'accès

4.1 Étude de l'existant

Contexte général : par demande d'accès, il faut entendre aussi bien la demande de création d'un accès que la demande de consultation, de modification ou de suppression d'un accès existant

4.1.1 Décrivez la manière suivant laquelle vous recevez les demandes d'accès. Quels sont les différents moyens(quel media, quelle procédure) mis à la disposition du demandeur pour effectuer sa demande ?

4.1.2 Quelles sont les différentes étapes du traitement de la demande (de sa réception à sa mise en œuvre) ? Décrivez chaque étape et les différents acteurs qui interviennent.

4.1.3 Y a-t-il des points de synchronisation ou d'attente dans le traitement de la demande ? Si oui, lesquels ?

4.1.4 Quel est, selon vous, le délai moyen de service d'une demande d'accès (entre la réception de la demande et la confirmation/infirmation de la réalisation complète de celle-ci) ?

4.1.5 Le demandeur reçoit-il un feedback ? Dans quels cas ? (toujours ?) Si oui, quel en est le contenu ?

4.1.6 Archivez-vous les demandes d'accès (ne pas confondre avec l'inventaire des accès) ? Si oui, à quelles fins ? Pendant combien de temps ? Y a-t-il des contraintes légales ou organisationnelles qui imposent cela ? Si oui, quelles sont-elles ?

4.1.7 Si une demande d'accès concerne plusieurs ressources, la considérez-vous comme une seule ou bien comme plusieurs demandes ? En quoi son traitement diffère-t-il de celui que vous avez décrit ou point 4.1.2 ?

4.1.8 Si une demande d'accès concerne plusieurs personnes, la considérez-vous comme une seule ou bien comme plusieurs demandes ? En quoi son traitement diffère-t-il de celui que vous avez décrit ou point 4.1.2 ?

4.2 Besoins et attentes

4.2.1 Quelles fonctionnalités principales attendriez-vous d'un système de gestion centralisée des accès ? Classez-les par ordre décroissant d'importance à vos yeux.

4.2.2 Quelles qualités principales attendriez-vous d'un système de gestion centralisée des accès ? Classez-les par ordre décroissant d'importance à vos yeux.

4.2.3 Quels avantages ou bénéfices estimeriez-vous devoir retirer d'un système de gestion centralisée des accès ?

4.2.4 Existe-t-il dans votre environnement, à votre connaissance, un système (ou un projet) dédicacé à cette problématique ? Si oui, lequel ? Peut-on en obtenir les spécifications ?

4.2.5 Quelles fonctionnalités de consultation (= requêtes) devrait selon vous offrir le système ?

5 Questions pour les demandeurs d'accès

5.1 Étude de l'existant

Contexte général : par demande d'accès, il faut entendre aussi bien la demande de création d'un accès que la demande de consultation, de modification ou de suppression d'un accès existant

5.1.1 Décrivez la manière suivant laquelle vous soumettez une demande d'accès. Quels sont les différents moyens(quel media, quelle procédure) mis à votre disposition pour effectuer cette demande ?

5.1.2 Quelles sont, à votre connaissance, les différentes étapes du traitement de votre demande (de sa réception à sa mise en œuvre) ? Décrivez chaque étape et les différents acteurs qui, selon vous, interviennent.

5.1.3 Y a-t-il, à votre connaissance, des points de synchronisation ou d'attente dans le traitement de votre demande ? Si oui, lesquels ?

5.1.4 Quel est, selon vous, le délai moyen de service d'une demande d'accès (entre la réception de la demande et la confirmation/infirmation de la réalisation complète de celle-ci) ?

5.1.5 Recevez-vous un feedback ? Dans quels cas ? (toujours ?) si oui, quel en est le contenu ?

5.1.6 Si une demande d'accès concerne plusieurs ressources, devez-vous faire plusieurs demandes ?

5.1.7 Si une demande d'accès concerne plusieurs personnes, devez-vous faire plusieurs demandes ?

5.2 Besoins et attentes

5.2.1 quelles fonctionnalités principales attendriez-vous d'un système de gestion centralisée des accès ? classez-les par ordre décroissant d'importance à vos yeux.

5.2.2 quelles qualités principales attendriez-vous d'un système de gestion centralisée des accès ? classez-les par ordre décroissant d'importance à vos yeux.

5.2.3 Quels avantages ou bénéfices estimeriez-vous devoir retirer d'un système centralisé de gestion des accès ?

5.2.4 Quelles fonctionnalités de consultation (= requêtes) devrait selon vous offrir le système ?

5.2.5 Existe-t-il dans votre environnement, à votre connaissance, un système (ou un projet) dédicacé à cette problématique ? si oui, lequel ? peut-on en obtenir les spécifications ?

6 Questions pour les gestionnaires de ressource

6.1 Étude de l'existant

Contexte général : par demande d'accès, il faut entendre aussi bien la demande de création d'un accès que la demande de consultation, de modification ou de suppression d'un accès existant.

6.1.1 Décrivez la manière suivant laquelle vous recevez les demandes d'accès aux ressources que vous gérez. Quels sont, dans votre contexte, les différents moyens (quel media, quelle procédure) mis à la disposition du demandeur pour effectuer sa demande.

6.1.2 Quelles sont, pour vous, les différentes étapes du traitement de la demande (de sa réception à sa mise en œuvre) ? Décrivez chaque étape et les différents acteurs qui interviennent.

6.1.3 Y a-t-il des points de synchronisation ou d'attente dans le traitement de la demande ? si oui, lesquels ?

6.1.4 Quel est, selon vous, le délai moyen de service d'une demande d'accès (entre la réception de la demande et la confirmation/infirmation de la réalisation complète de celle-ci) ?

6.1.5 Donnez-vous un feedback sur le traitement de la demande ou son état d'avancement ? à qui ? dans quels cas ? (toujours ?) quel en est le contenu ?

6.1.6 Archivez-vous les demandes d'accès (ne pas confondre avec l'inventaire des accès) ? si oui, à quelles fins ? pendant combien de temps ? Y a-t-il des contraintes légales ou organisationnelles qui imposent cela ? si oui, quelles sont-elles ?

6.1.7 Quelles ressources (type, nature) gérez-vous actuellement ?

6.1.8 Si une demande d'accès concerne plusieurs ressources, la considérez-vous comme une seule ou bien comme plusieurs demandes ? en quoi son traitement diffère-t-il de celui que vous avez décrit ou point 6.1.2 ?

6.1.9 Si une demande d'accès concerne plusieurs personnes, la considérez-vous comme une seule ou bien comme plusieurs demandes ? en quoi son traitement diffère-t-il de celui que vous avez décrit ou point 6.1.2 ?

6.2 Besoins et attentes

6.2.1 quelles fonctionnalités principales attendriez-vous d'un système de gestion centralisée des accès ? classez-les par ordre décroissant d'importance à vos yeux.

6.2.2 quelles qualités principales attendriez-vous d'un système de gestion centralisée des accès ? classez-les par ordre décroissant d'importance à vos yeux.

6.2.3 Quels avantages ou bénéfices estimeriez-vous devoir retirer d'un système centralisé de gestion des accès ?

6.2.4 Existe-t-il dans votre environnement, à votre connaissance, un système (ou un projet) dédicacé à cette problématique ? si oui, lequel ? peut-on en obtenir les spécifications ?

6.2.5 Quel rôle souhaiteriez-vous jouer dans un tel système ?

6.2.6 Quelles fonctionnalités de consultation (= requêtes) devrait selon vous offrir le système ?